King County Renovation: Most Effective Ways to Increase the Value of Your Home

Author: E. Berke Tezcan

TABLE OF CONTENTS

Click to jump to matching Markdown Header.

- **INTRODUCTION**
- **OBTAIN**
- **SCRUB/EXPLORE**
- **MODEL**
- **iNTERPRET**
- **CONCLUSIONS/RECOMMENDATIONS**

INTRODUCTION

For this project, we were hired by a home owner in King County, Washington who wants to renovate their home. They would like us to analyze the real estate data of the county and give them insights as to what to focus their renovation efforts on in order to increase their property's value.

Real estate prices are affected by a myriad of -what we can define as- internal parameters like the square footage, floor count, bedroom count, finishes, how many cars can fit into the garage etc. There are also parameters that we can define as external and can not (easily) be changed. These include attributes like zipcode, latitude, longitude, view from the house, and school districts. In order to accurately model and pinpoint the most important parameters that affect the sale price of a home, we need to incorporate both internal and external parameters.

We are given a dataset that includes information about the real estate in King County and will be using this dataset to create a multiple linear regression (MLR) model. We defined our goal to be to find the top 3 internal parameters that affect a home's sale price the most in King County specifically. This will ensure that the home owner can actually keep these parameters in mind while renovating rather than getting insights about external parameters that they can't necessarily do anything to change.

OBTAIN

Data Understanding/EDA

In [1]:
```
import pandas as pd
import seaborn as sns
```

```
import matplotlib.pyplot as plt
# from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
import statsmodels.api as sm
import statsmodels.stats.api as sms
import statsmodels.formula.api as smf
import numpy as np
from scipy import stats
```

In [2]:
```
df = pd.read_csv('data/kc_house_data.csv')
```

In [3]:
```
pd.set_option('display.max_columns',0)
df.head()
```

Out[3]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	vi
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650	1.0	NaN	
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242	2.0	0.0	
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000	1.0	0.0	
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000	1.0	0.0	
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080	1.0	0.0	

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             21597 non-null  int64  
 1   date           21597 non-null  object 
 2   price          21597 non-null  float64
 3   bedrooms       21597 non-null  int64  
 4   bathrooms      21597 non-null  float64
 5   sqft_living    21597 non-null  int64  
 6   sqft_lot       21597 non-null  int64  
 7   floors         21597 non-null  float64
 8   waterfront     19221 non-null  float64
 9   view           21534 non-null  float64
 10  condition      21597 non-null  int64  
 11  grade          21597 non-null  int64  
 12  sqft_above     21597 non-null  int64  
 13  sqft_basement  21597 non-null  object 
 14  yr_built       21597 non-null  int64  
 15  yr_renovated   17755 non-null  float64
 16  zipcode        21597 non-null  int64  
 17  lat            21597 non-null  float64
 18  long           21597 non-null  float64
 19  sqft_living15  21597 non-null  int64  
 20  sqft_lot15     21597 non-null  int64  
dtypes: float64(8), int64(11), object(2)
memory usage: 3.5+ MB
```

In [5]:
```
df['bathrooms'].unique()
```

Out[5]:
```
array([1.  , 2.25, 3.  , 2.  , 4.5 , 1.5 , 2.5 , 1.75, 2.75, 3.25, 4.  ,
       3.5 , 0.75, 4.75, 5.  , 4.25, 3.75, 1.25, 5.25, 6.  , 0.5 , 5.5 ,
```

```
       6.75, 5.75, 8.  , 7.5 , 7.75, 6.25, 6.5 ])
```

In [6]:
```
df['condition'].unique()
```

Out[6]:
```
array([3, 5, 4, 1, 2], dtype=int64)
```

In [7]:
```
df['grade'].unique()
```

Out[7]:
```
array([ 7,  6,  8, 11,  9,  5, 10, 12,  4,  3, 13], dtype=int64)
```

In [8]:
```
df['view'].unique()
```

Out[8]:
```
array([ 0., nan,  3.,  4.,  2.,  1.])
```

In [9]:
```
pd.DataFrame(df['id'].unique())
```

Out[9]:

	0
0	7129300520
1	6414100192
2	5631500400
3	2487200875
4	1954400510
...	...
21415	263000018
21416	6600060120
21417	1523300141
21418	291310100
21419	1523300157

21420 rows × 1 columns

We have 21,597 individual home sales associated with 21,420 unique homes.

SCRUB/EXPLORE

Addressing Null Values

In [10]:
```
df.isna().sum()
```

Out[10]:
```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
```

```
waterfront       2376
view               63
condition           0
grade               0
sqft_above          0
sqft_basement       0
yr_built            0
yr_renovated     3842
zipcode             0
lat                 0
long                0
sqft_living15       0
sqft_lot15          0
dtype: int64
```

To address the missing values in the view column we can take the more conservative approach and say that these 63 houses/apartments did not have a view with minimal impact to the overall dataset since we have 21597 data points overall.

In [11]:
```
df['view'].fillna(0, inplace=True)
df.isna().sum()
```

Out[11]:
```
id                  0
date                0
price               0
bedrooms            0
bathrooms           0
sqft_living         0
sqft_lot            0
floors              0
waterfront       2376
view                0
condition           0
grade               0
sqft_above          0
sqft_basement       0
yr_built            0
yr_renovated     3842
zipcode             0
lat                 0
long                0
sqft_living15       0
sqft_lot15          0
dtype: int64
```

In [12]:
```
df[df['yr_built']==df['yr_renovated']]
```

Out[12]:

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grac

There are no oddities with the year built and year renovated columns so far it seems like.

In [13]:
```
df[df['yr_renovated'].isna()]
```

Out[13]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfro
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000	1.0	(
12	114101516	5/28/2014	310000.0	3	1.00	1430	19901	1.5	(
23	8091400200	5/16/2014	252700.0	2	1.50	1070	9643	1.0	Nɛ

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfro
26	1794500383	6/26/2014	937000.0	3	1.75	2450	2691	2.0	(
28	5101402488	6/24/2014	438000.0	3	1.75	1520	6380	1.0	(
...	...	...	...	...	...	...	...	...	
21576	1931300412	4/16/2015	475000.0	3	2.25	1190	1200	3.0	(
21577	8672200110	3/17/2015	1090000.0	5	3.75	4170	8142	2.0	(
21579	1972201967	10/31/2014	520000.0	2	2.25	1530	981	3.0	(
21581	191100405	4/21/2015	1580000.0	4	3.25	3410	10125	2.0	(
21583	7202300110	9/15/2014	810000.0	4	3.00	3990	7838	2.0	(

3842 rows × 21 columns

In [14]:
```
df['waterfront'].unique()
```

Out[14]:
```
array([nan,  0.,  1.])
```

In [15]:
```
df['yr_renovated'].unique()
```

Out[15]:
```
array([   0., 1991.,   nan, 2002., 2010., 1992., 2013., 1994., 1978.,
       2005., 2003., 1984., 1954., 2014., 2011., 1983., 1945., 1990.,
       1988., 1977., 1981., 1995., 2000., 1999., 1998., 1970., 1989.,
       2004., 1986., 2007., 1987., 2006., 1985., 2001., 1980., 1971.,
       1979., 1997., 1950., 1969., 1948., 2009., 2015., 1974., 2008.,
       1968., 2012., 1963., 1951., 1962., 1953., 1993., 1996., 1955.,
       1982., 1956., 1940., 1976., 1946., 1975., 1964., 1973., 1957.,
       1959., 1960., 1967., 1965., 1934., 1972., 1944., 1958.])
```

I assumed here that the NA values meant that the place was not renovated. This is also the more conservative approach so I'm comfortable with replacing these values with 0.

In [16]:
```
df['yr_renovated'].fillna(0, inplace=True)
```

In [17]:
```
df.isna().sum()
```

Out[17]:
```
id                  0
date                0
price               0
bedrooms            0
bathrooms           0
sqft_living         0
sqft_lot            0
floors              0
waterfront       2376
view                0
condition           0
grade               0
sqft_above          0
sqft_basement       0
yr_built            0
yr_renovated        0
zipcode             0
lat                 0
```

```
long              0
sqft_living15     0
sqft_lot15        0
dtype: int64
```

Similar to other columns, putting 0's for the NaN values in the waterfront column is the more conservative approach and makes sense.

In [18]:
```
df['waterfront'].fillna(0, inplace=True)
```

In [19]:
```
df.isna().sum()
```

Out[19]:
```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
```

Addressing Placeholder Values

In [20]:
```
df['sqft_basement'].unique()
```

Out[20]:
```
array(['0.0', '400.0', '910.0', '1530.0', '?', '730.0', '1700.0', '300.0',
       '970.0', '760.0', '720.0', '700.0', '820.0', '780.0', '790.0',
       '330.0', '1620.0', '360.0', '588.0', '1510.0', '410.0', '990.0',
       '600.0', '560.0', '550.0', '1000.0', '1600.0', '500.0', '1040.0',
       '880.0', '1010.0', '240.0', '265.0', '290.0', '800.0', '540.0',
       '710.0', '840.0', '380.0', '770.0', '480.0', '570.0', '1490.0',
       '620.0', '1250.0', '1270.0', '120.0', '650.0', '180.0', '1130.0',
       '450.0', '1640.0', '1460.0', '1020.0', '1030.0', '750.0', '640.0',
       '1070.0', '490.0', '1310.0', '630.0', '2000.0', '390.0', '430.0',
       '850.0', '210.0', '1430.0', '1950.0', '440.0', '220.0', '1160.0',
       '860.0', '580.0', '2060.0', '1820.0', '1180.0', '200.0', '1150.0',
       '1200.0', '680.0', '530.0', '1450.0', '1170.0', '1080.0', '960.0',
       '280.0', '870.0', '1100.0', '460.0', '1400.0', '660.0', '1220.0',
       '900.0', '420.0', '1580.0', '1380.0', '475.0', '690.0', '270.0',
       '350.0', '935.0', '1370.0', '980.0', '1470.0', '160.0', '950.0',
       '50.0', '740.0', '1780.0', '1900.0', '340.0', '470.0', '370.0',
       '140.0', '1760.0', '130.0', '520.0', '890.0', '1110.0', '150.0',
       '1720.0', '810.0', '190.0', '1290.0', '670.0', '1800.0', '1120.0',
       '1810.0', '60.0', '1050.0', '940.0', '310.0', '930.0', '1390.0',
       '610.0', '1830.0', '1300.0', '510.0', '1330.0', '1590.0', '920.0',
       '1320.0', '1420.0', '1240.0', '1960.0', '1560.0', '2020.0',
       '1190.0', '2110.0', '1280.0', '250.0', '2390.0', '1230.0', '170.0',
       '830.0', '1260.0', '1410.0', '1340.0', '590.0', '1500.0', '1140.0',
```

```
       '260.0', '100.0', '320.0', '1480.0', '1060.0', '1284.0', '1670.0',
       '1350.0', '2570.0', '1090.0', '110.0', '2500.0', '90.0', '1940.0',
       '1550.0', '2350.0', '2490.0', '1481.0', '1360.0', '1135.0',
       '1520.0', '1850.0', '1660.0', '2130.0', '2600.0', '1690.0',
       '243.0', '1210.0', '1024.0', '1798.0', '1610.0', '1440.0',
       '1570.0', '1650.0', '704.0', '1910.0', '1630.0', '2360.0',
       '1852.0', '2090.0', '2400.0', '1790.0', '2150.0', '230.0', '70.0',
       '1680.0', '2100.0', '3000.0', '1870.0', '1710.0', '2030.0',
       '875.0', '1540.0', '2850.0', '2170.0', '506.0', '906.0', '145.0',
       '2040.0', '784.0', '1750.0', '374.0', '518.0', '2720.0', '2730.0',
       '1840.0', '3480.0', '2160.0', '1920.0', '2330.0', '1860.0',
       '2050.0', '4820.0', '1913.0', '80.0', '2010.0', '3260.0', '2200.0',
       '415.0', '1730.0', '652.0', '2196.0', '1930.0', '515.0', '40.0',
       '2080.0', '2580.0', '1548.0', '1740.0', '235.0', '861.0', '1890.0',
       '2220.0', '792.0', '2070.0', '4130.0', '2250.0', '2240.0',
       '1990.0', '768.0', '2550.0', '435.0', '1008.0', '2300.0', '2610.0',
       '666.0', '3500.0', '172.0', '1816.0', '2190.0', '1245.0', '1525.0',
       '1880.0', '862.0', '946.0', '1281.0', '414.0', '2180.0', '276.0',
       '1248.0', '602.0', '516.0', '176.0', '225.0', '1275.0', '266.0',
       '283.0', '65.0', '2310.0', '10.0', '1770.0', '2120.0', '295.0',
       '207.0', '915.0', '556.0', '417.0', '143.0', '508.0', '2810.0',
       '20.0', '274.0', '248.0'], dtype=object)
```

The sqft_basement column's dtype needs to be int like other sqft columns but it is dtype object due to the "?" values.

In [21]:

```
df[df['sqft_basement']=='?']
```

Out[21]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfro
6	1321400060	6/27/2014	257500.0	3	2.25	1715	6819	2.0	(
18	16000397	12/5/2014	189000.0	2	1.00	1200	9850	1.0	(
42	7203220400	7/7/2014	861990.0	5	2.75	3595	5639	2.0	(
79	1531000030	3/23/2015	720000.0	4	2.50	3450	39683	2.0	(
112	2525310310	9/16/2014	272500.0	3	1.75	1540	12600	1.0	(
...	...	...	...	...	...	...	...	...	
21442	3226049565	7/11/2014	504600.0	5	3.00	2360	5000	1.0	(
21447	1760650900	7/21/2014	337500.0	4	2.50	2330	4907	2.0	(
21473	6021503707	1/20/2015	352500.0	2	2.50	980	1010	3.0	(
21519	2909310100	10/15/2014	332000.0	4	2.50	2380	5737	2.0	(
21581	191100405	4/21/2015	1580000.0	4	3.25	3410	10125	2.0	(

454 rows × 21 columns

It seems like only 454 rows out of 21,597 have ? as their value. Once again being conservative I'm going to assume that these houses/apartments don't have a basement and replace the ? with 0's and change the dtype to int.

In [22]:

```
df['sqft_basement'] = df['sqft_basement'].map(lambda x: x.replace('?', '0') if x=='?' e
# df['sqft_basement'] = df['sqft_basement'].map(lambda x: int(float(x)))
df['sqft_basement'] = df['sqft_basement'].astype('float').astype('int64')
```

In [23]:
```
df['sqft_basement'].dtype
```

Out[23]:
```
dtype('int64')
```

In [24]:
```
df['renovated'] = df['yr_renovated']!=0
df['renovated'] = df['renovated'].astype('int')
df.head()
```

Out[24]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	vi
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650	1.0	0.0	
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242	2.0	0.0	
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000	1.0	0.0	
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000	1.0	0.0	
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080	1.0	0.0	

In [25]:
```
df['has_basement'] = df['sqft_basement']!=0
df['has_basement'] = df['has_basement'].astype('int')
df.head()
```

Out[25]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	vi
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650	1.0	0.0	
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242	2.0	0.0	
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000	1.0	0.0	
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000	1.0	0.0	
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080	1.0	0.0	

In [26]:
```
df[df['id'].duplicated()]
```

Out[26]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfro
94	6021501535	12/23/2014	700000.0	3	1.50	1580	5000	1.0	(
314	4139480200	12/9/2014	1400000.0	4	3.25	4290	12103	1.0	(
325	7520000520	3/11/2015	240500.0	2	1.00	1240	12092	1.0	(
346	3969300030	12/29/2014	239900.0	4	1.00	1000	7134	1.0	(
372	2231500030	3/24/2015	530000.0	4	2.25	2180	10754	1.0	(
...	...	...	...	...	...	...	...	...	
20165	7853400250	2/19/2015	645000.0	4	3.50	2910	5260	2.0	(
20597	2724049222	12/1/2014	220000.0	2	2.50	1000	1092	2.0	(
20654	8564860270	3/30/2015	502000.0	4	2.50	2680	5539	2.0	(
20764	6300000226	5/4/2015	380000.0	4	1.00	1200	2171	1.5	(

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfro
21565	7853420110	5/4/2015	625000.0	3	3.00	2780	6000	2.0	(

177 rows × 23 columns

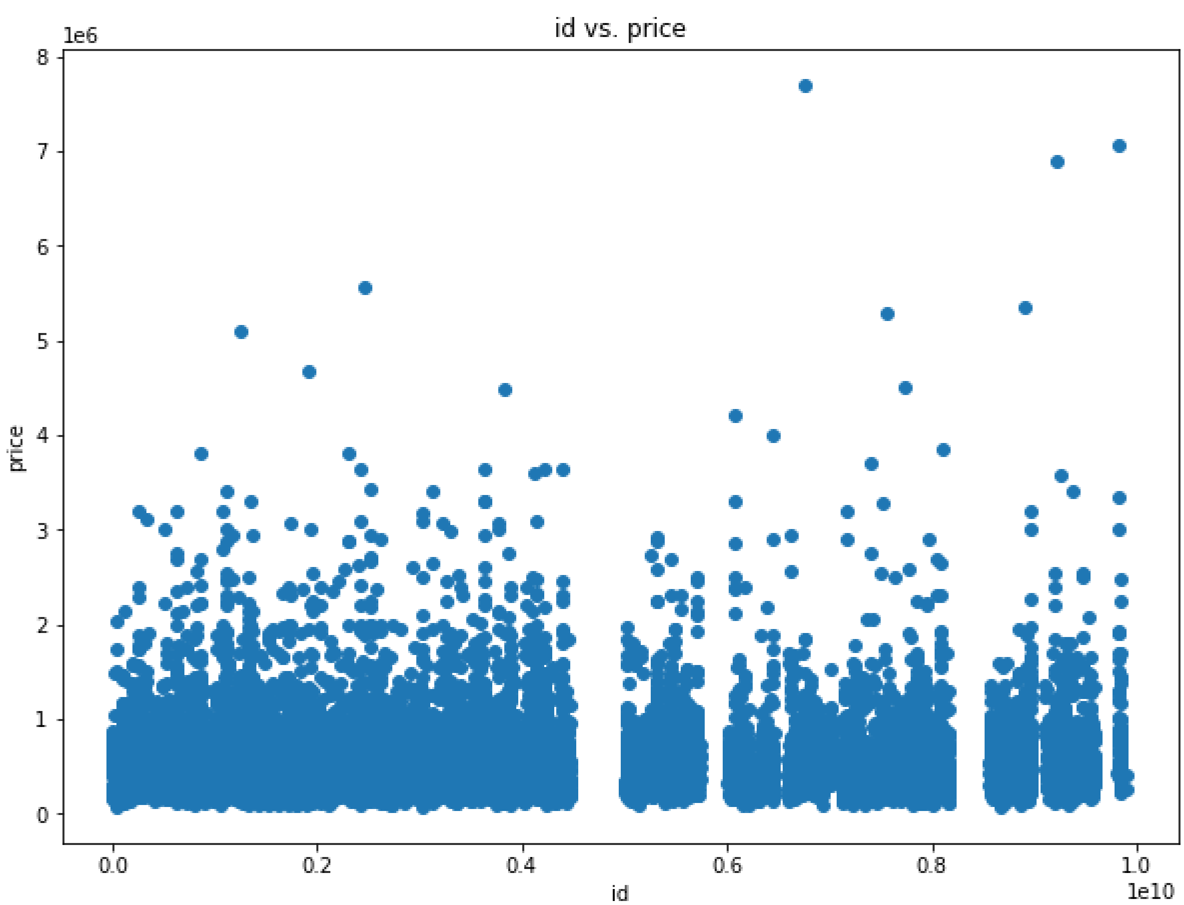
There are duplicated entries for the same houses but since these are additional sales of the same house, they are valid data points and there is no reason to drop them.

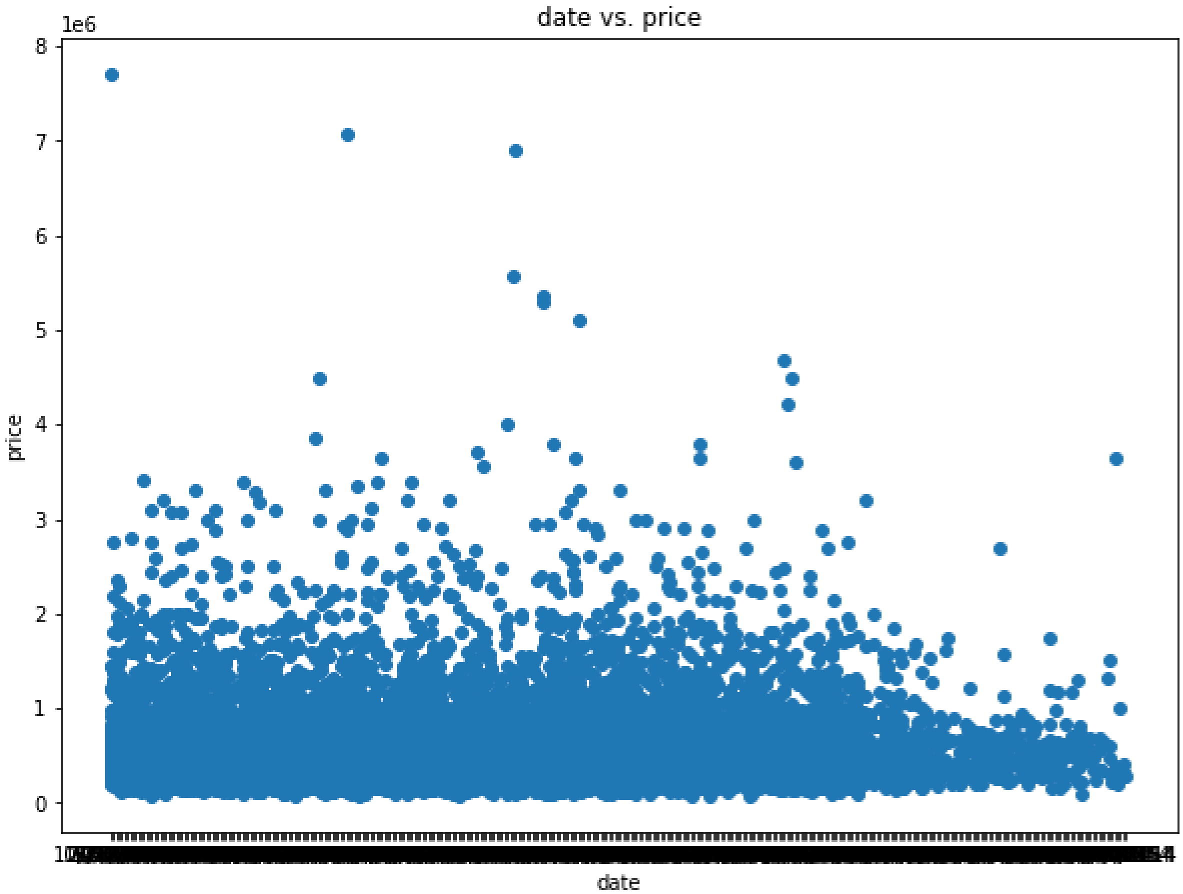
Linearity Check

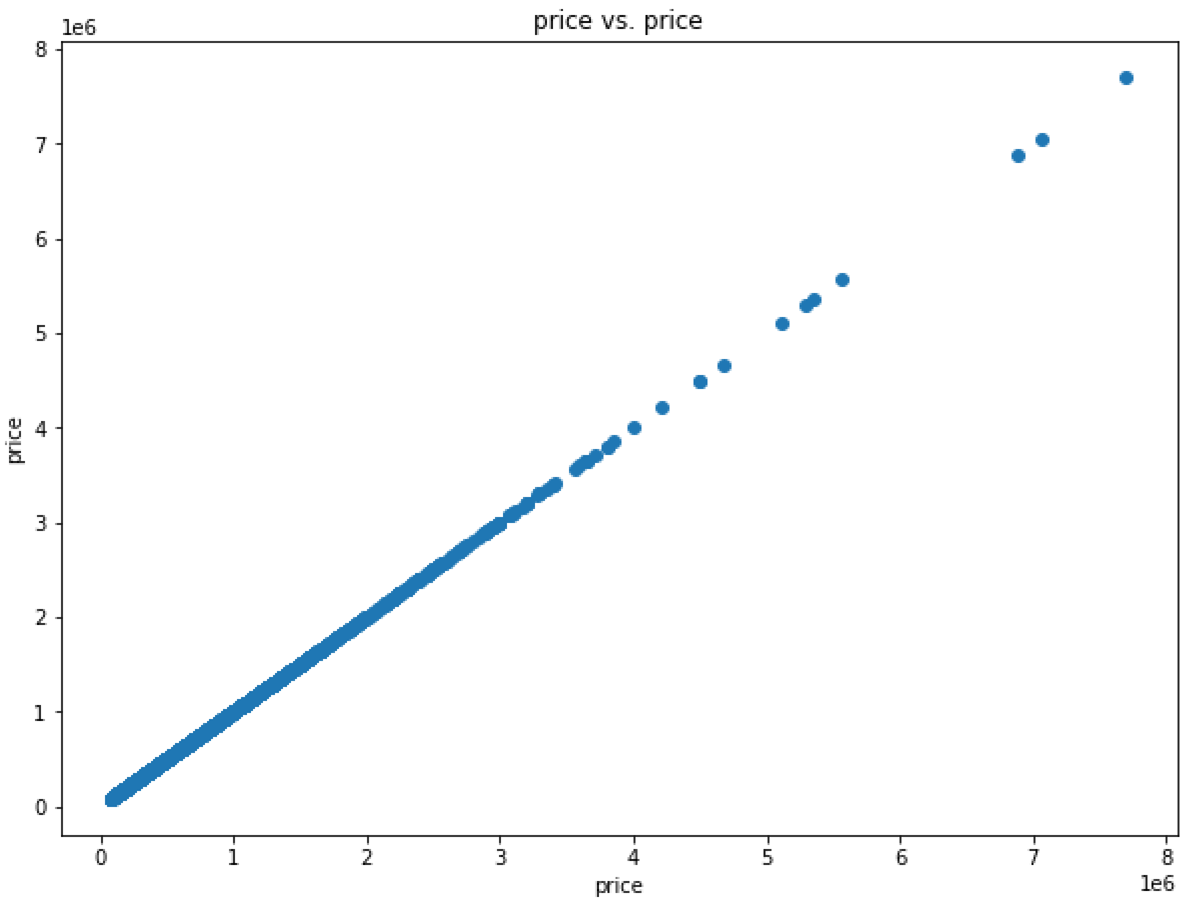
Checking for Linearity of Parameters

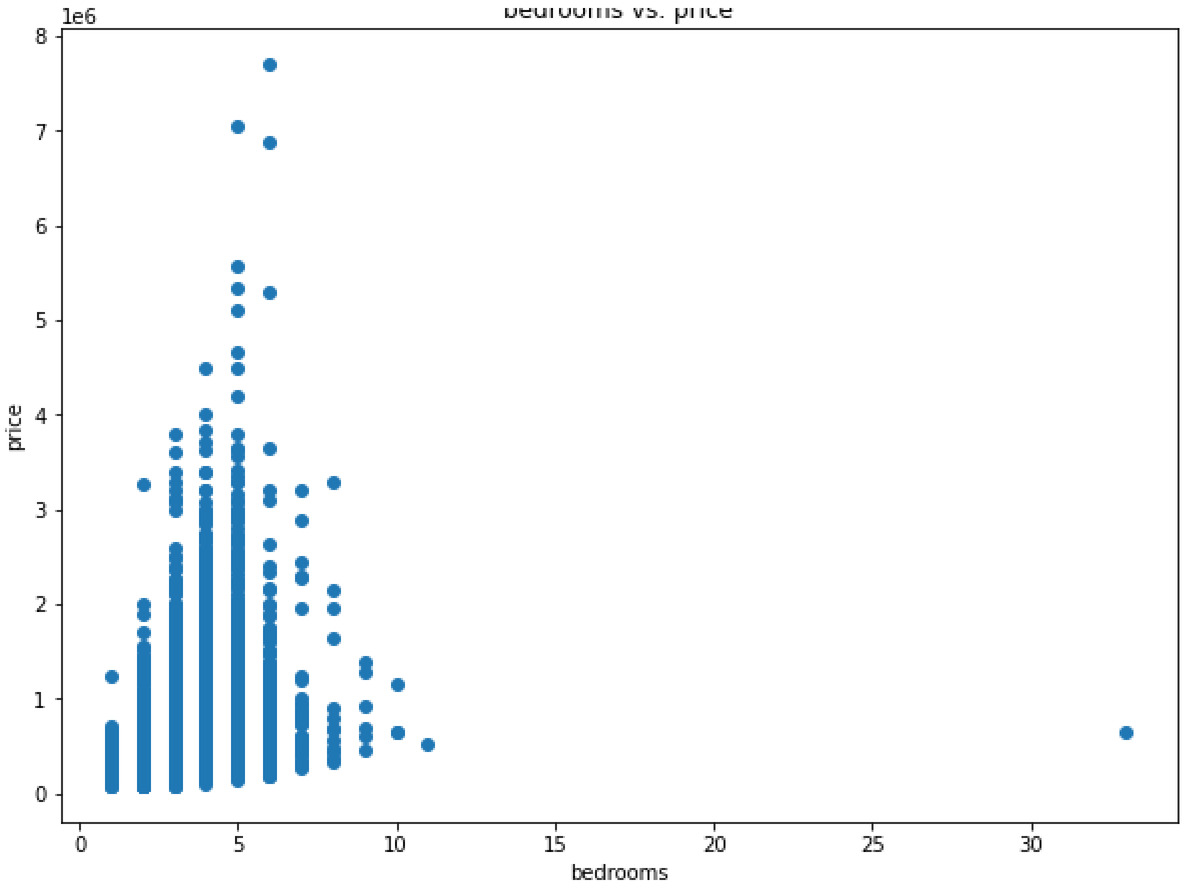
In [27]:
```
import seaborn as sns

def plot(df, target='price'):
    fig, ax = plt.subplots(nrows = len(df.columns), figsize=(10,200))

    for i, col in enumerate(df.columns):
#         sns.lmplot(x=col, y=target, data=df)
        ax[i].scatter(df[col], df[target])
        ax[i].set_xlabel(col)
        ax[i].set_ylabel(target)
        ax[i].set_title(f"{col} vs. {target}")
```

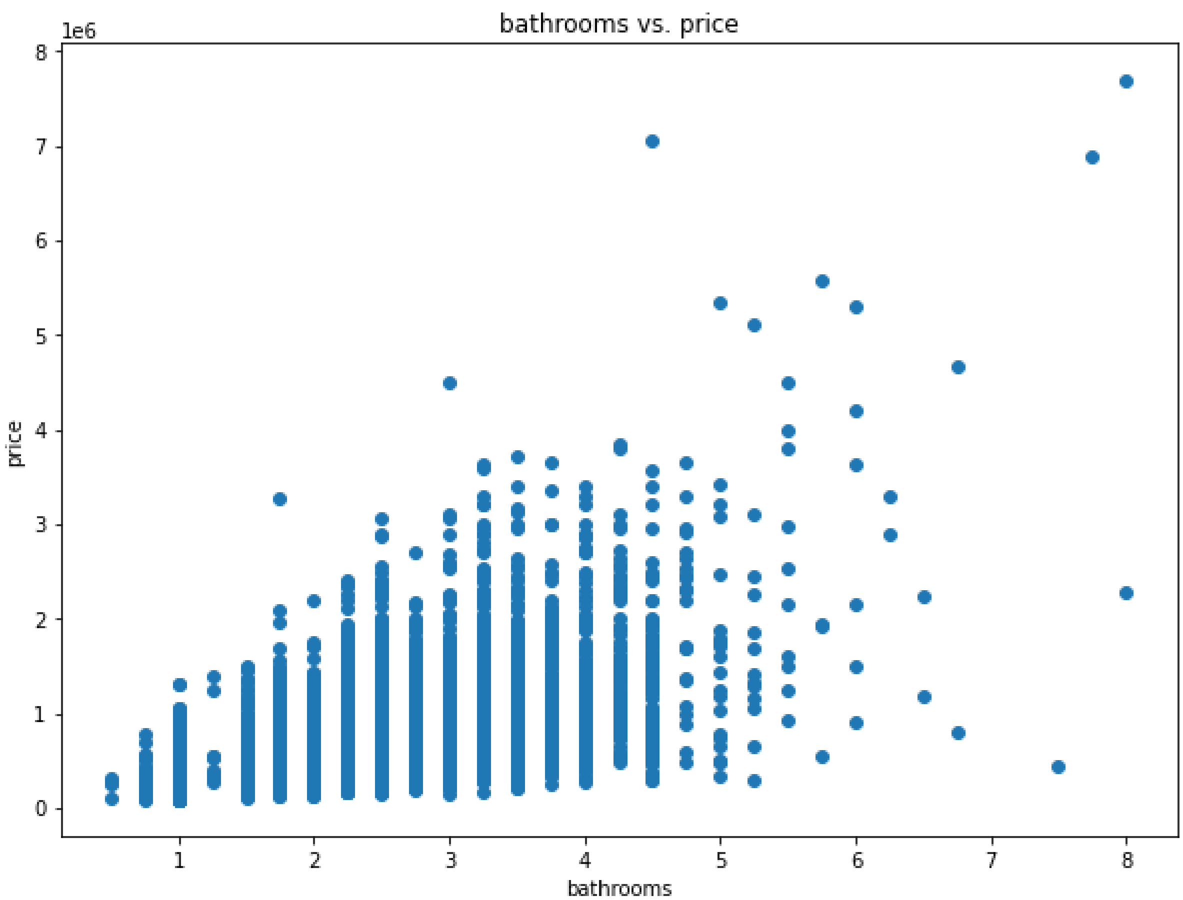
In [28]:
```
plot(df=df, target='price')
```

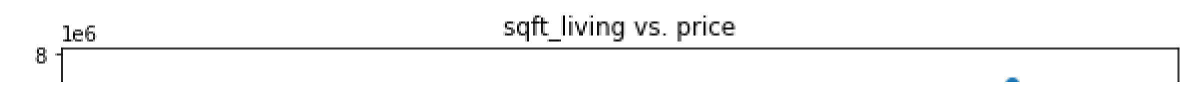


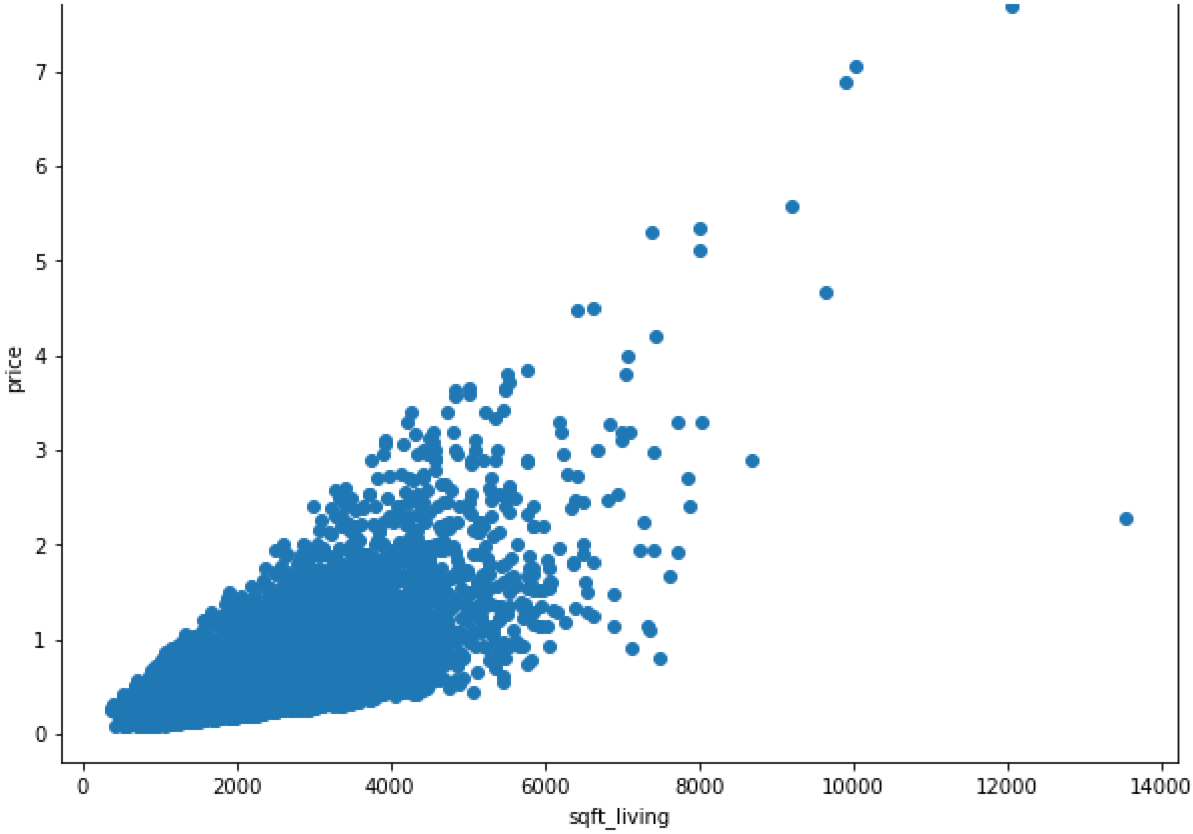

date vs. price
1e6
price
date

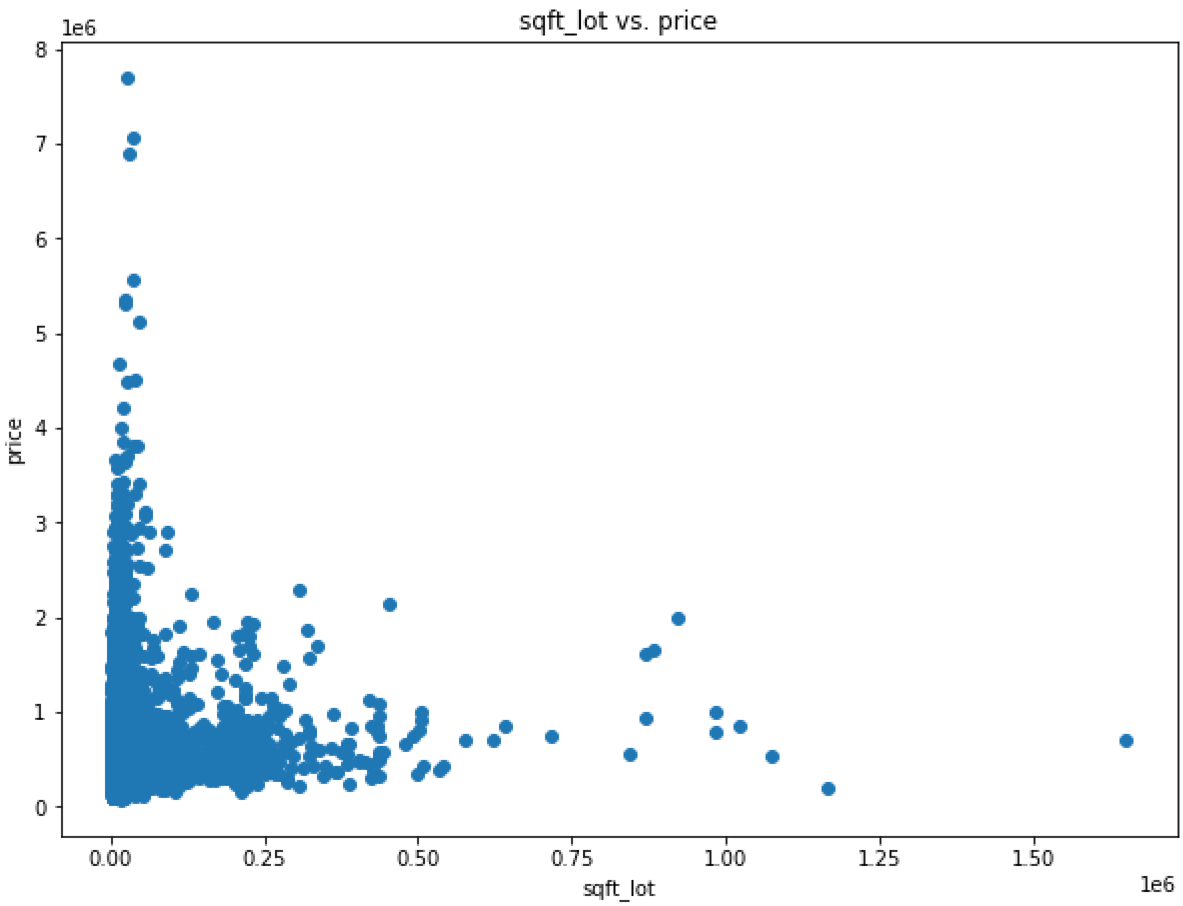

price vs. price
1e6
price
price
1e6

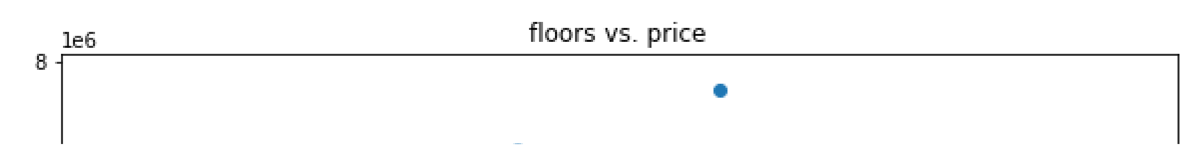

bedrooms vs. price
1e6
price
bedrooms

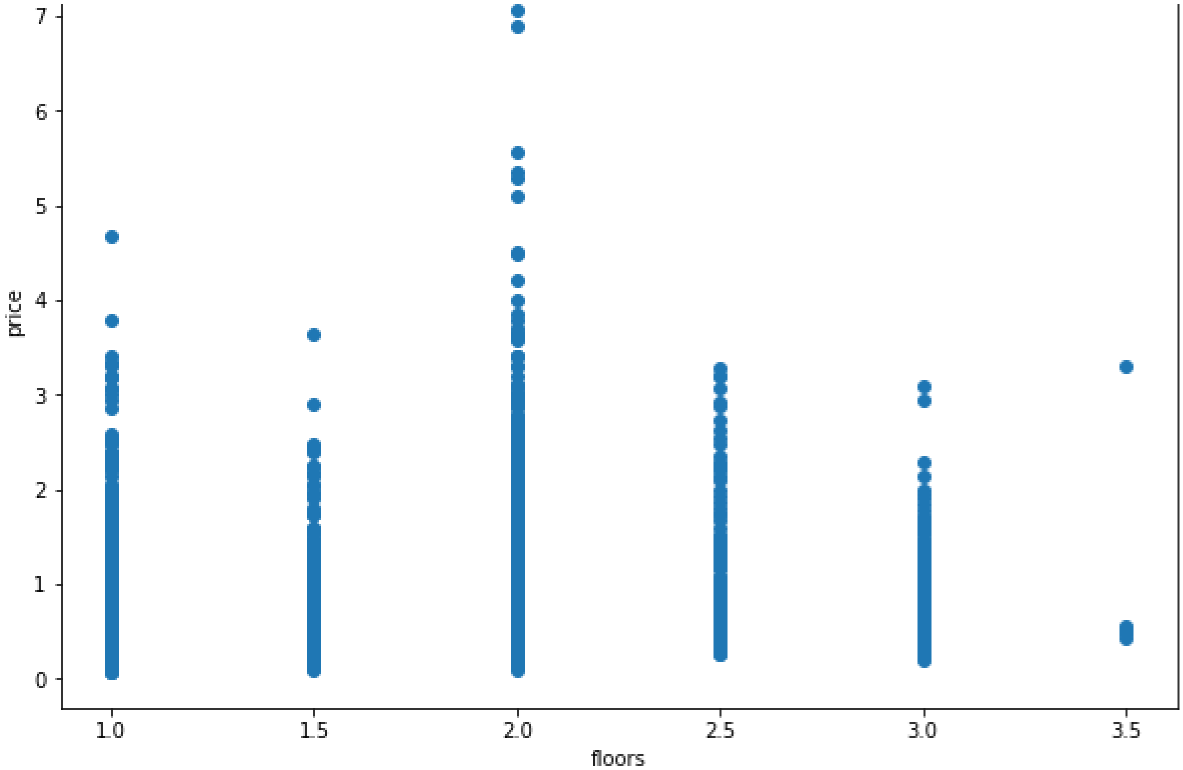

bathrooms vs. price
1e6
price
bathrooms

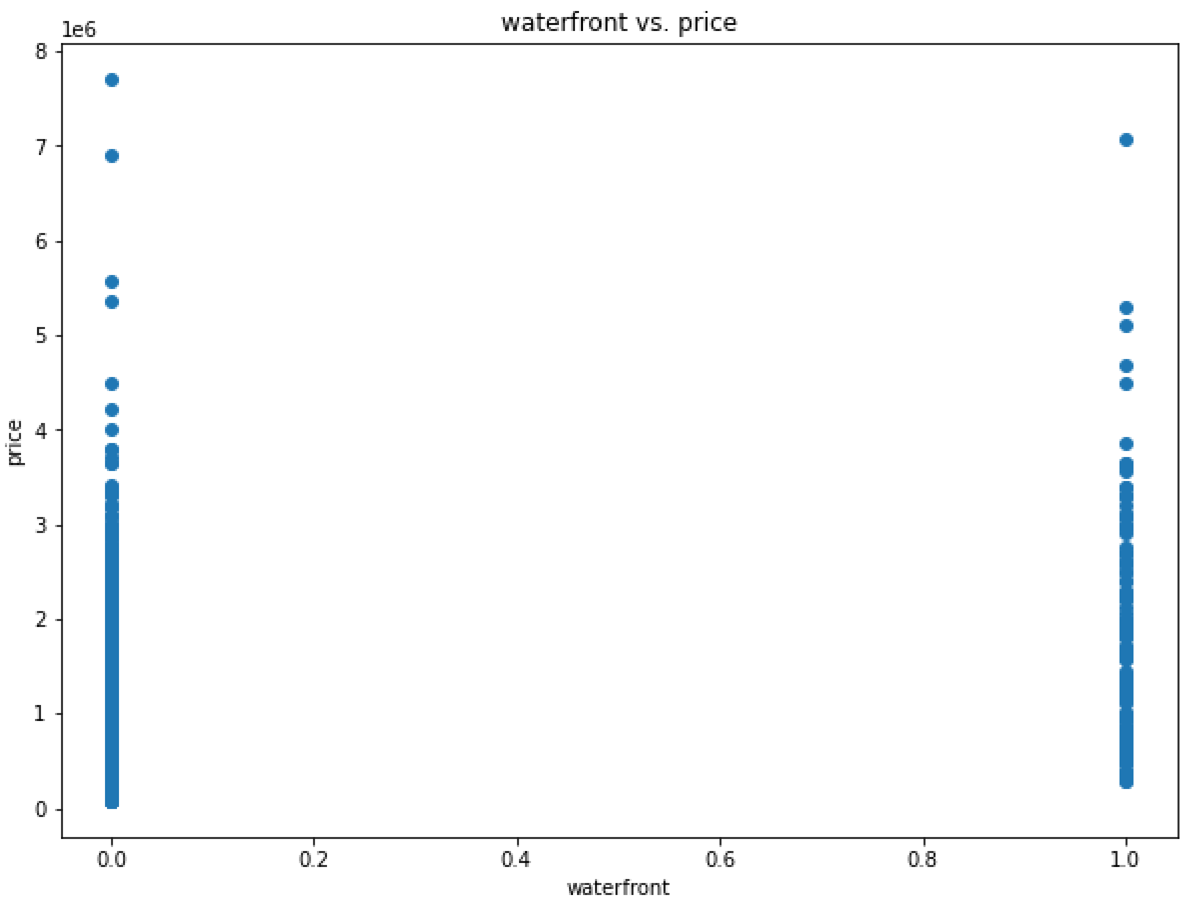

sqft_living vs. price
1e6

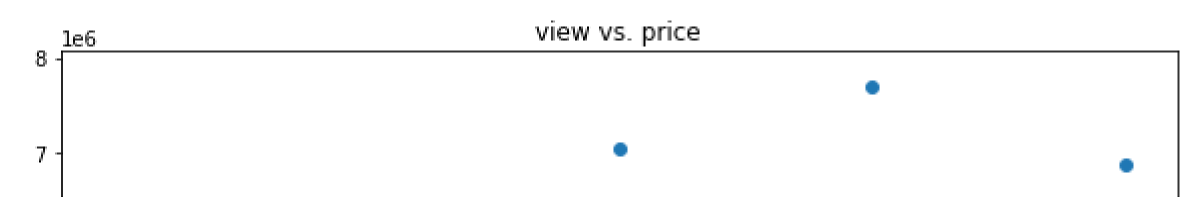

price
sqft_living
0
1
2
3
4
5
6
7
0
2000
4000
6000
8000
10000
12000
14000

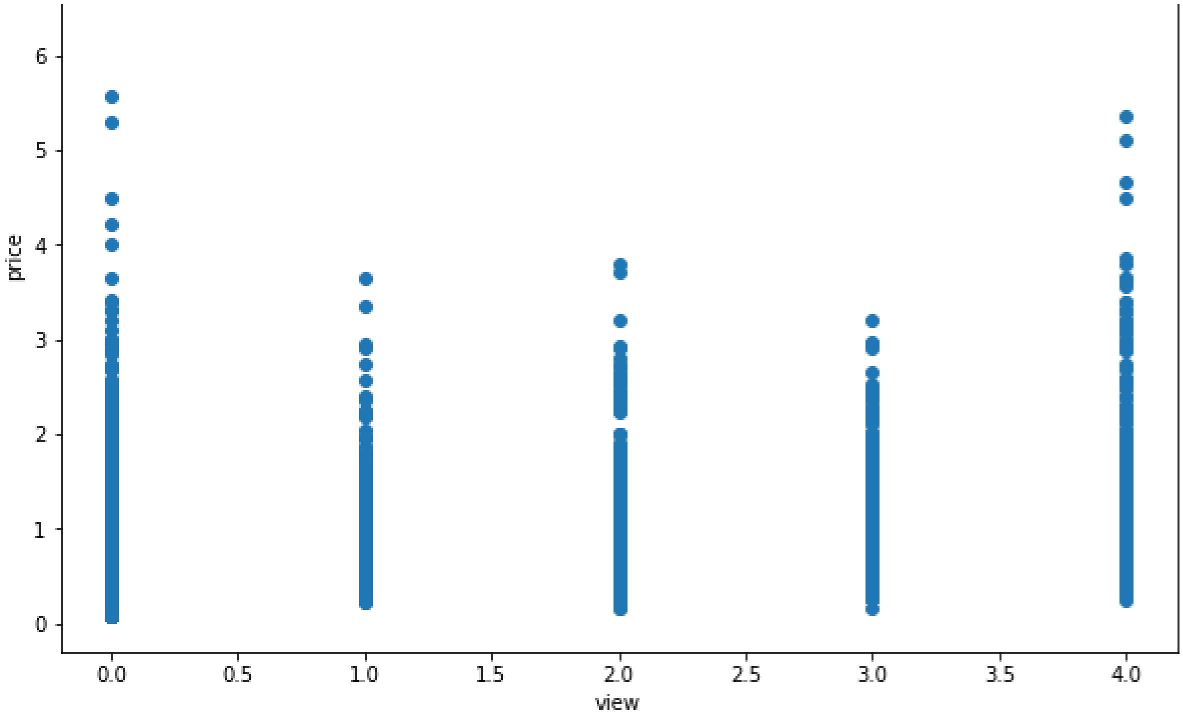

sqft_lot vs. price
1e6
price
sqft_lot
1e6
0
1
2
3
4
5
6
7
8
0.00
0.25
0.50
0.75
1.00
1.25
1.50


floors vs. price
1e6
8

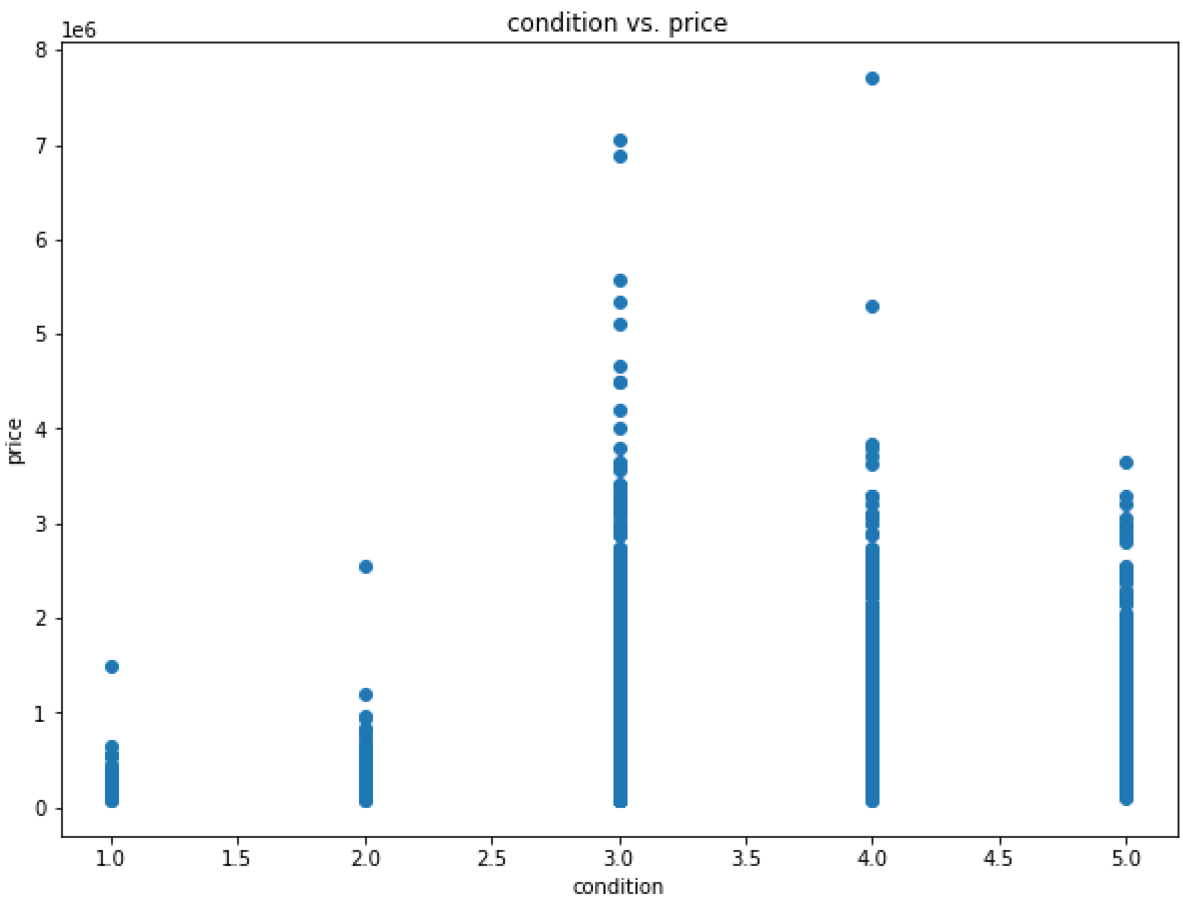

price
floors
1.0
1.5
2.0
2.5
3.0
3.5


waterfront vs. price
1e6
price
waterfront

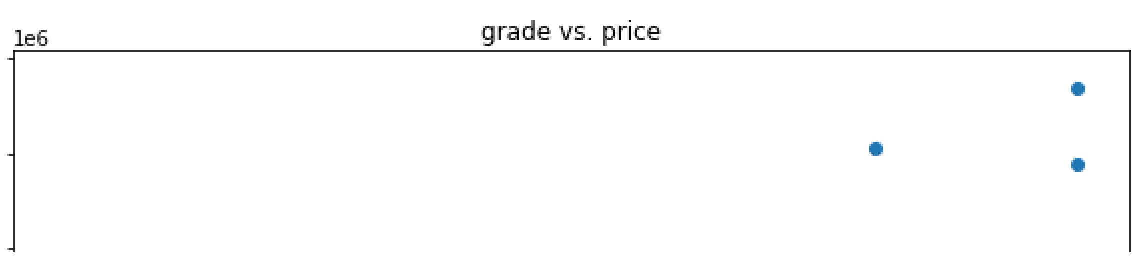

view vs. price
1e6



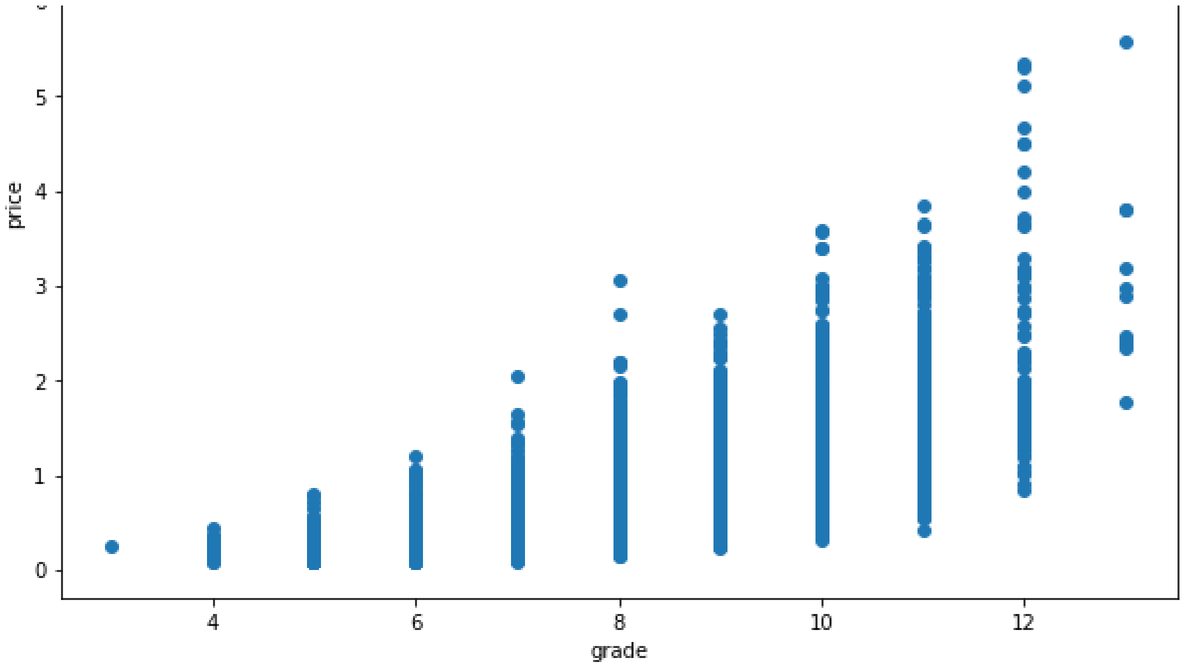
price
view
0
1
2
3
4
5
6
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0



condition vs. price
1e6
price
condition
0
1
2
3
4
5
6
7
8
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

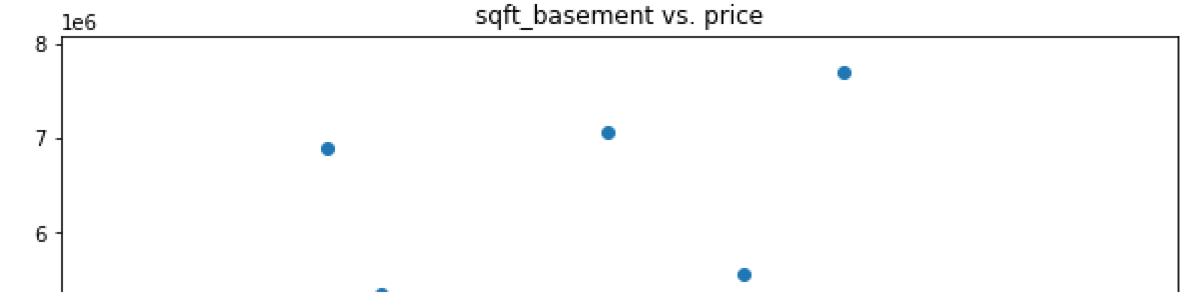


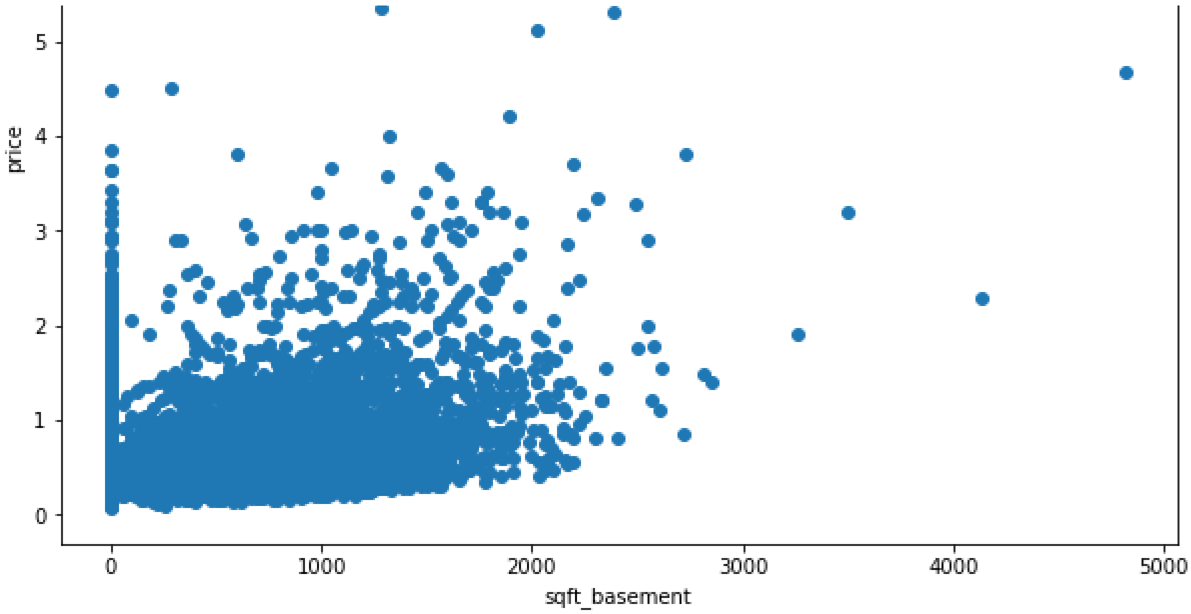
grade vs. price
1e6
8
7
6

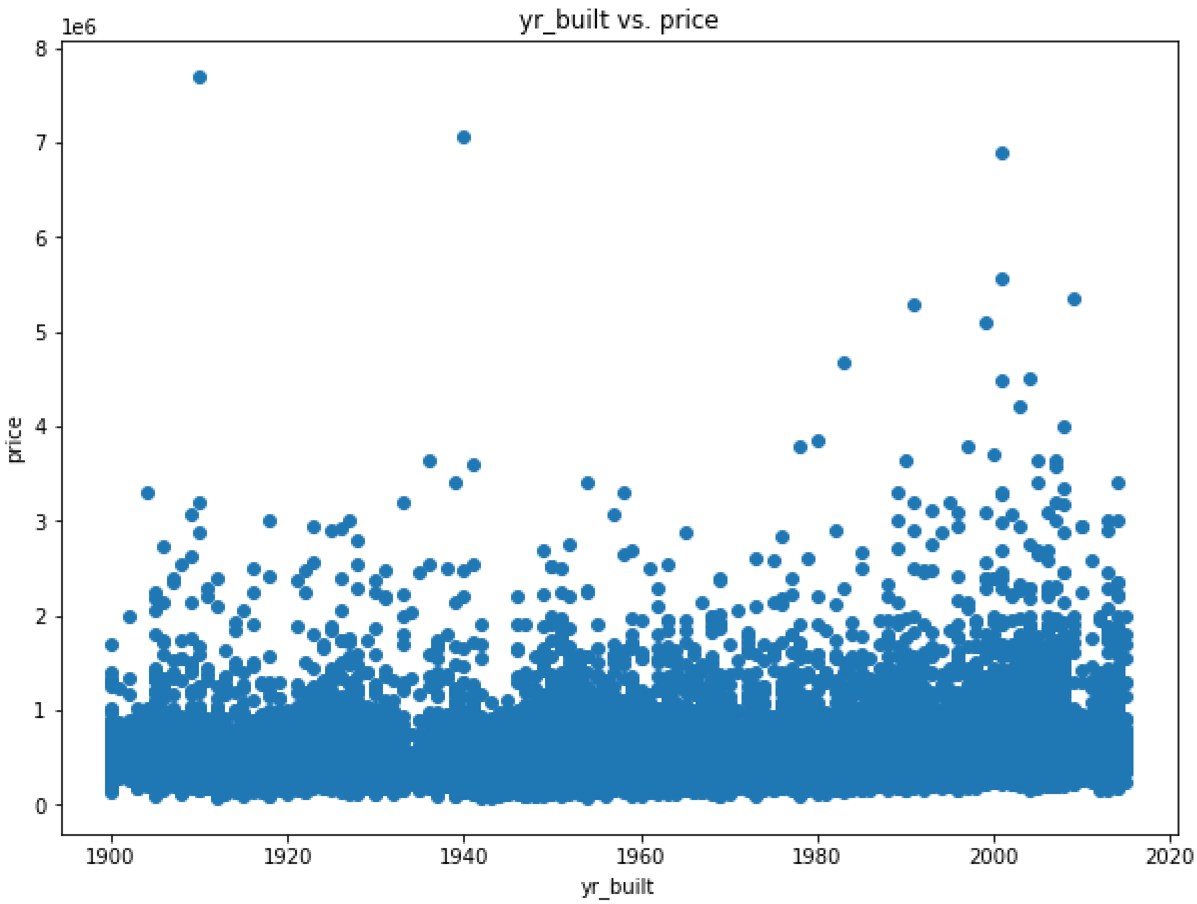


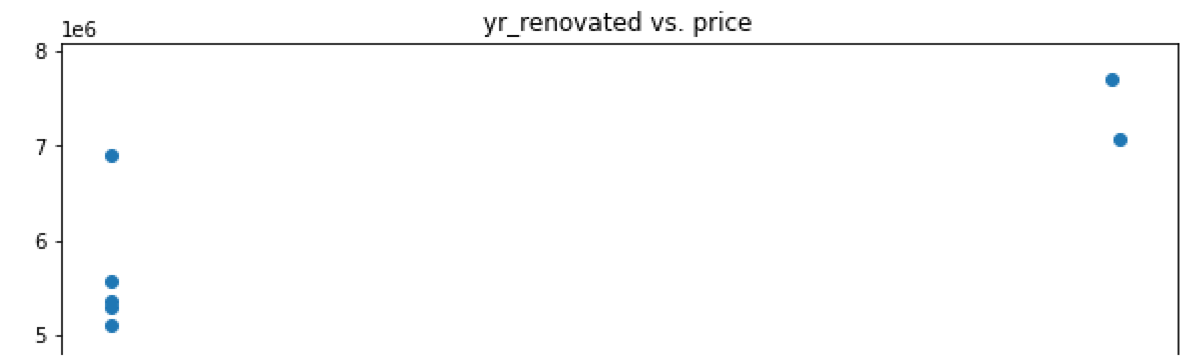
sqft_above vs. price

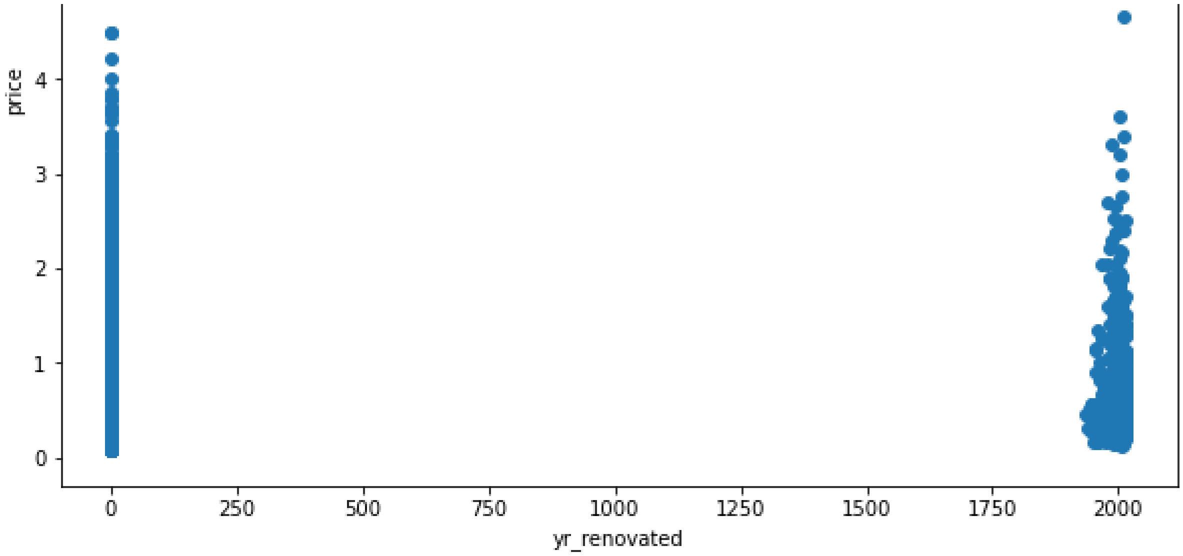
sqft_basement vs. price

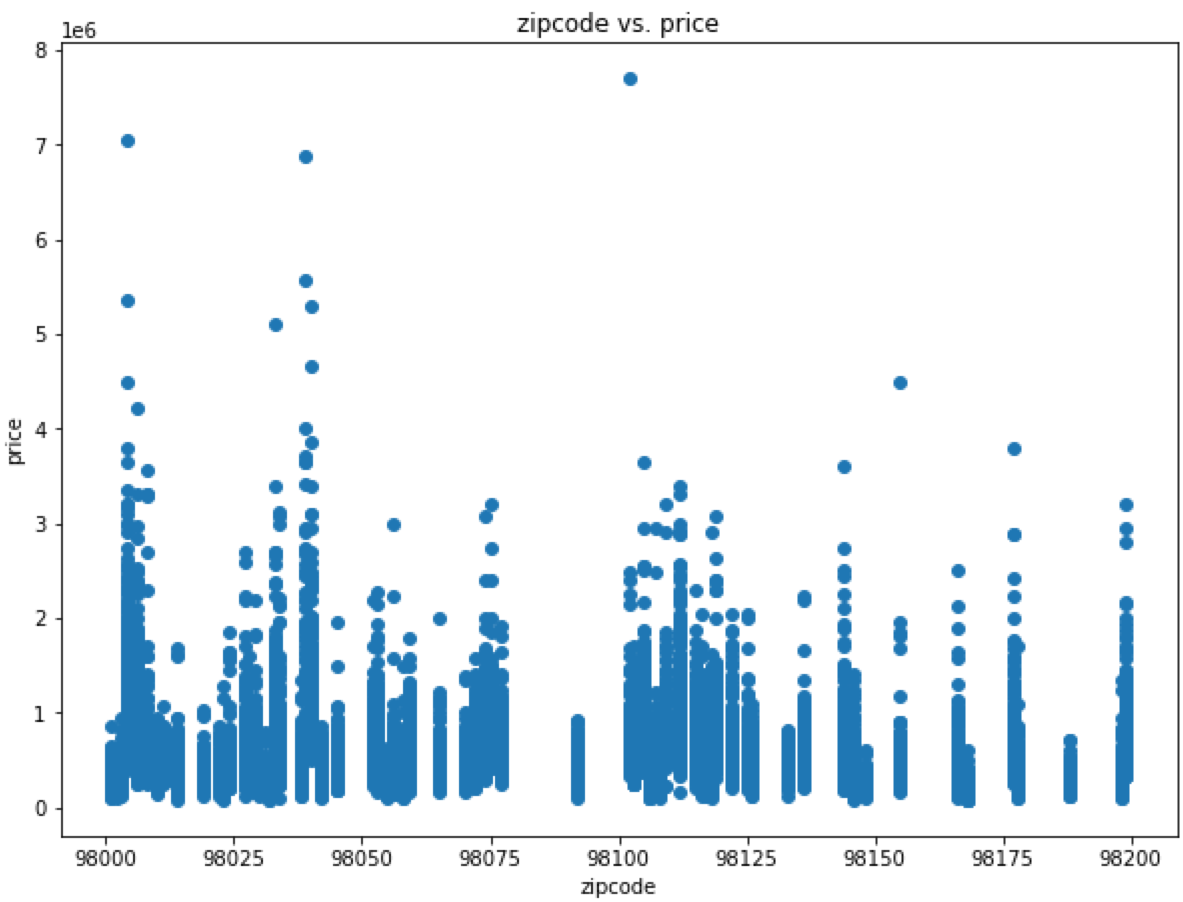


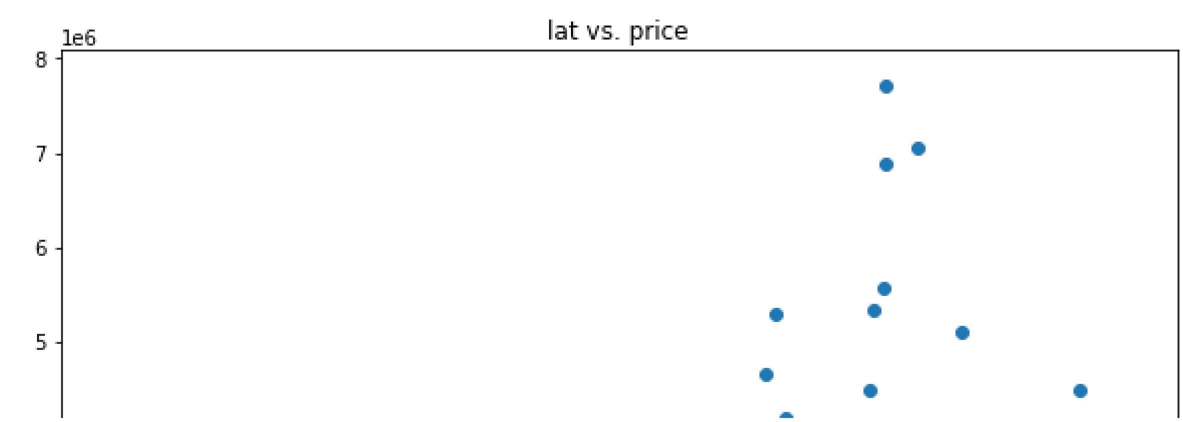

price
sqft_basement
0
1
2
3
4
5
1000
2000
3000
4000
5000

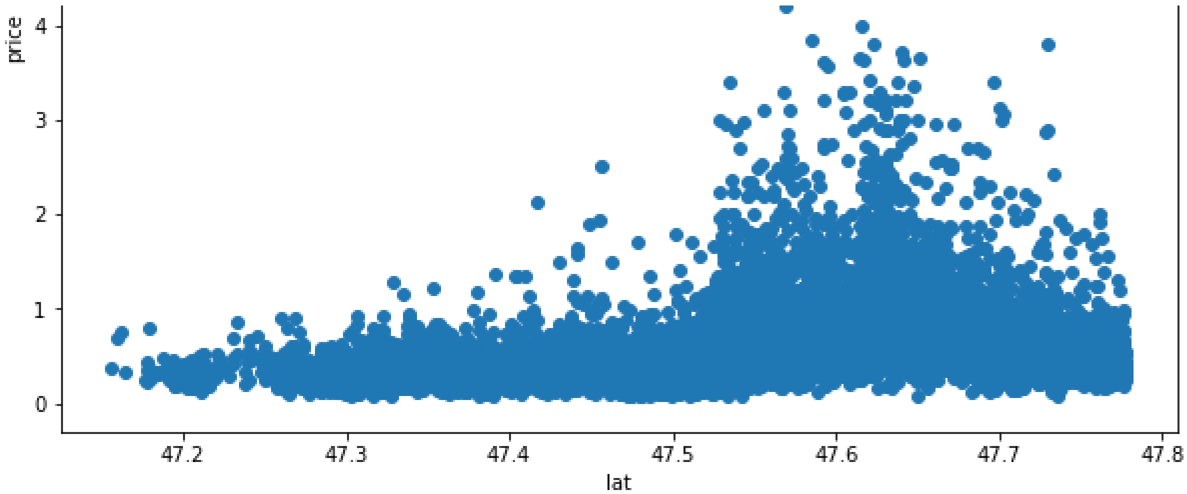

yr_built vs. price
1e6
price
yr_built
0
1
2
3
4
5
6
7
8
1900
1920
1940
1960
1980
2000
2020

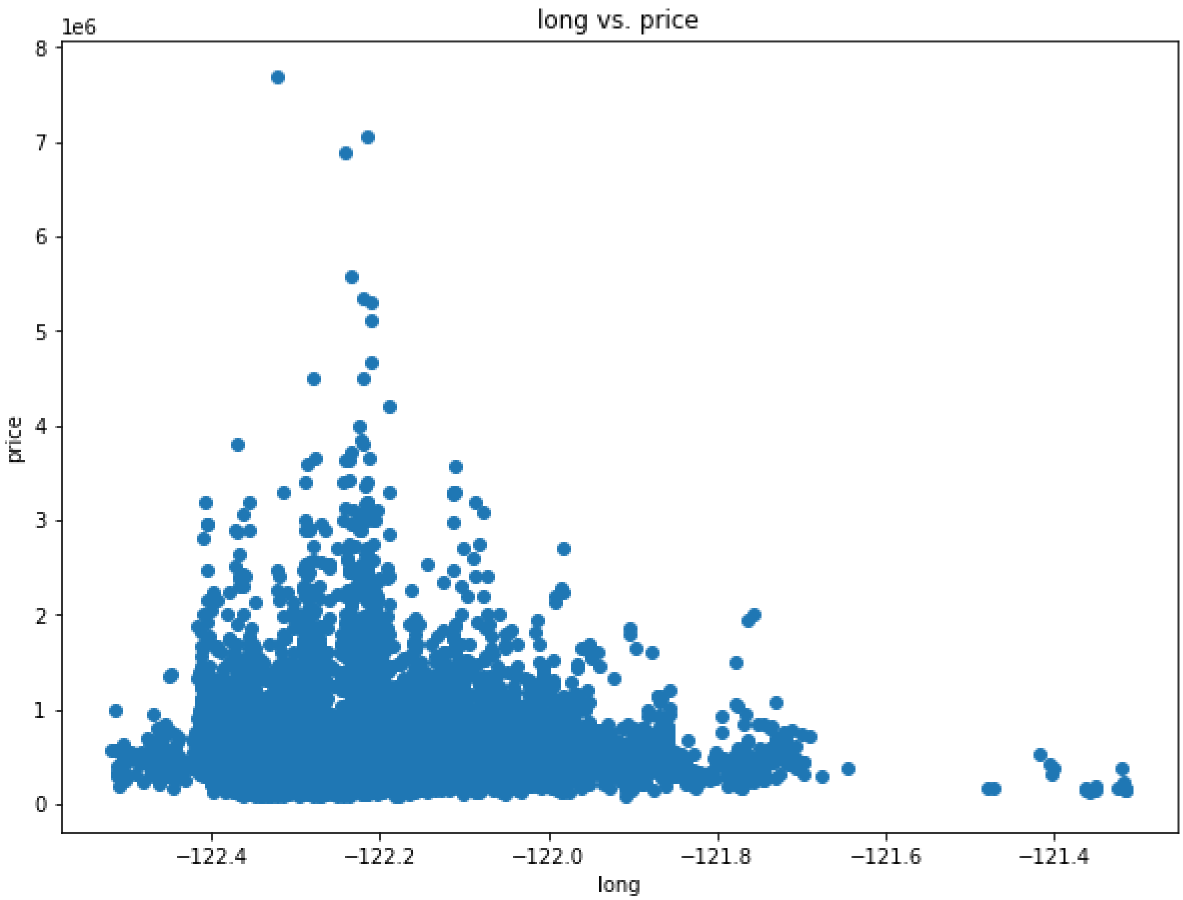

yr_renovated vs. price
1e6
5
6
7
8

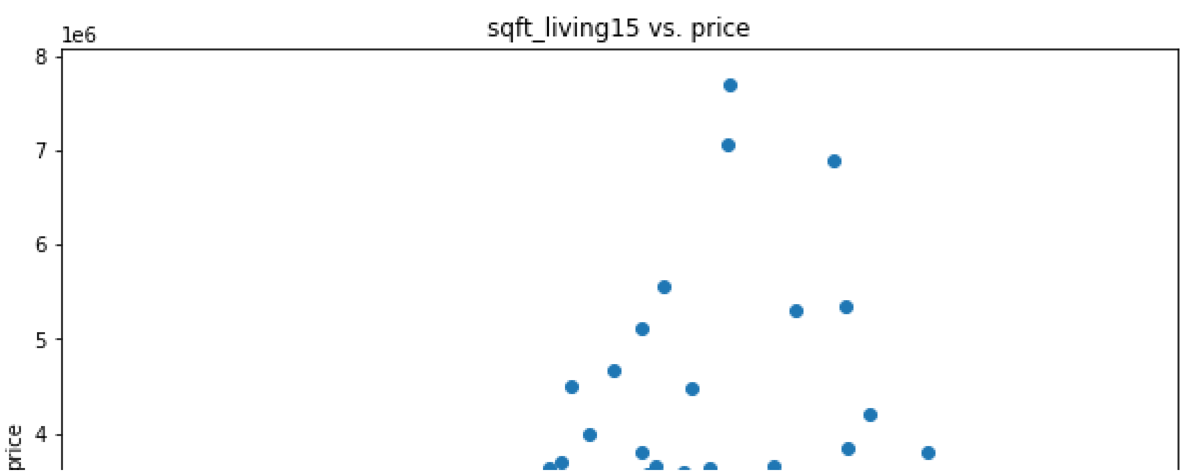

price
yr_renovated
0
250
500
750
1000
1250
1500
1750
2000
0
1
2
3
4

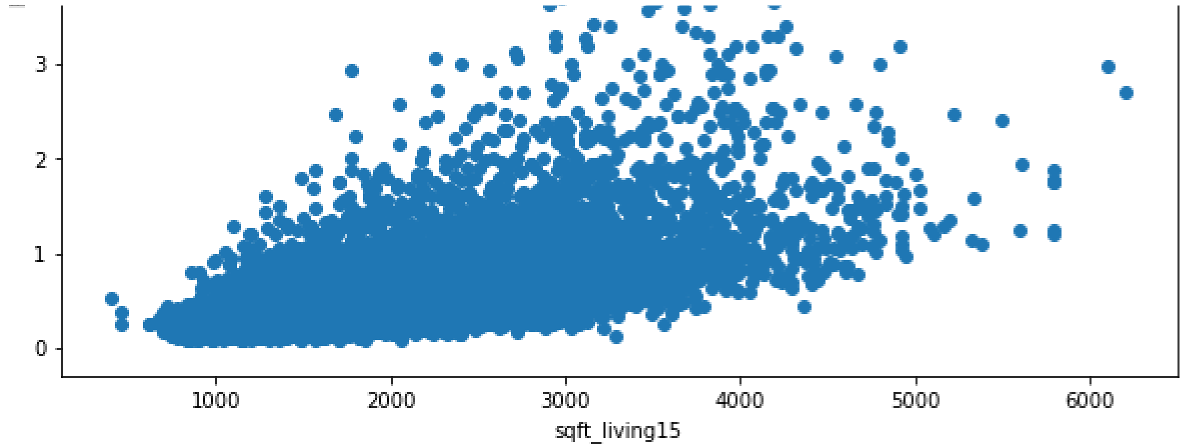

zipcode vs. price
1e6
price
zipcode
98000
98025
98050
98075
98100
98125
98150
98175
98200
0
1
2
3
4
5
6
7
8

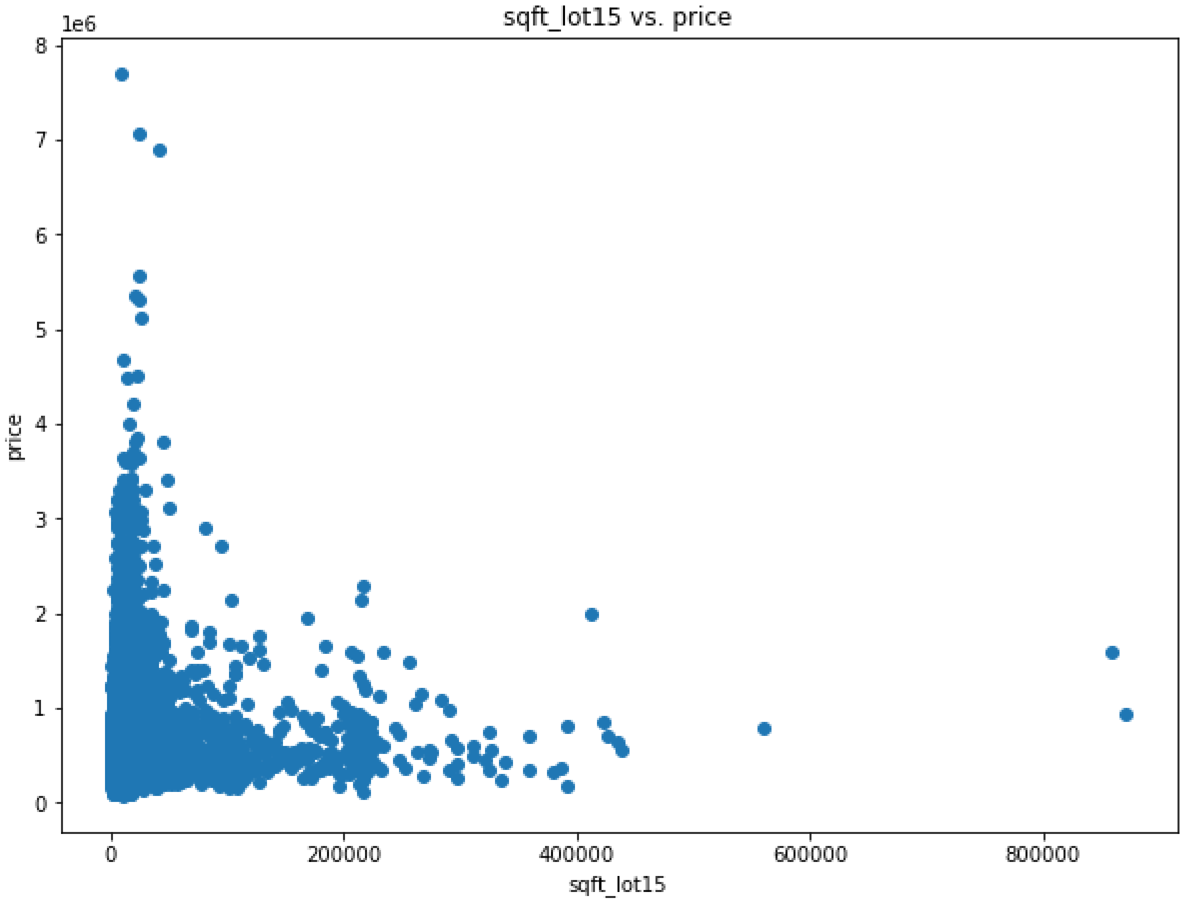

lat vs. price
1e6
5
6
7
8


price
4
3
2
1
0
47.2
47.3
47.4
47.5
47.6
47.7
47.8
lat

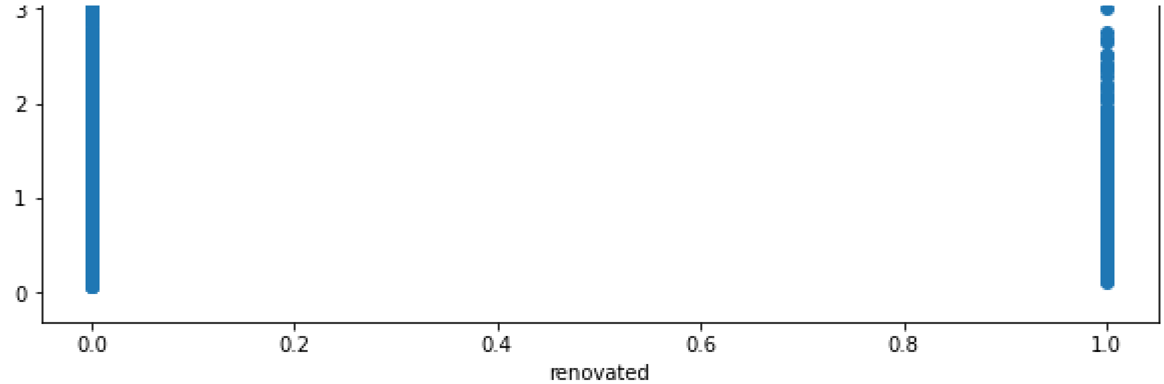

long vs. price
1e6
8
7
6
5
4
3
2
1
0
price
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
long

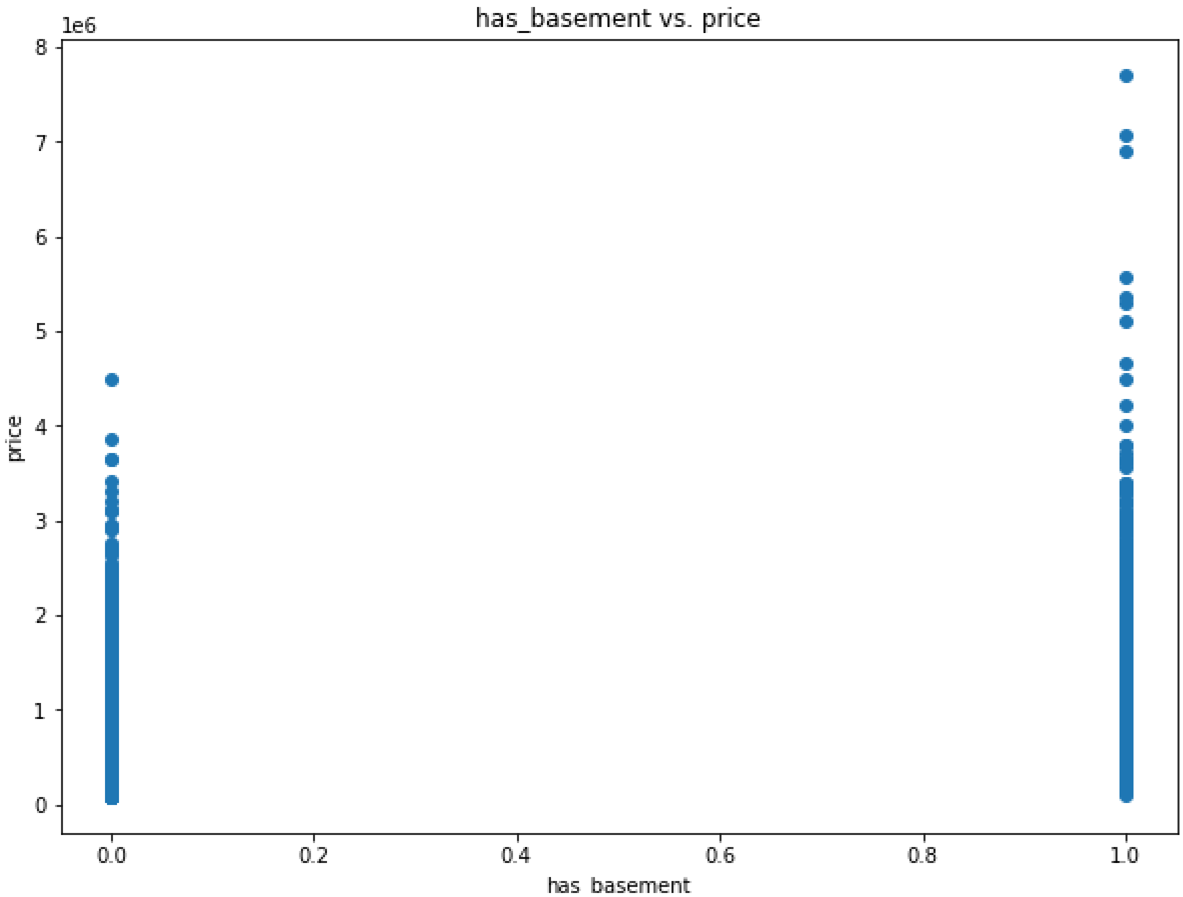

sqft_living15 vs. price
1e6
8
7
6
5
4
price


0
1
2
3
1000
2000
3000
4000
5000
6000
sqft_living15

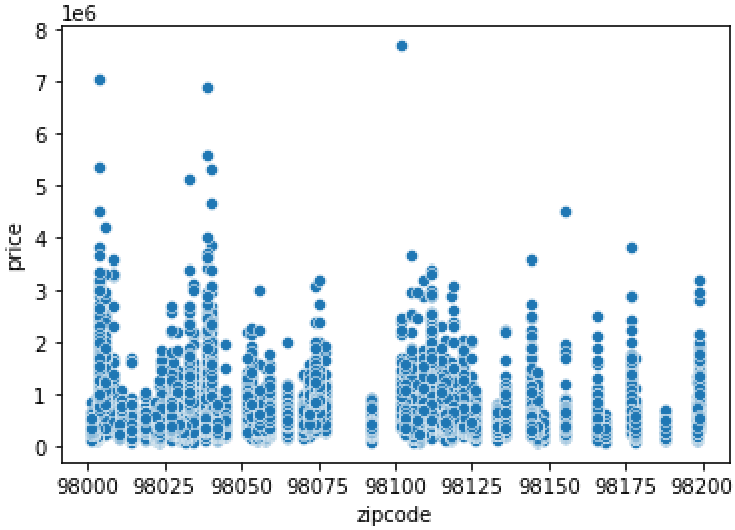

sqft_lot15 vs. price
1e6
price
0
1
2
3
4
5
6
7
8
0
200000
400000
600000
800000
sqft_lot15


renovated vs. price
1e6
price
4
5
6
7
8





In [29]:
```
sns.scatterplot(x=df['zipcode'], y=df['price']);
```



Zipcode is clearly a categorical column with no linear relationship with our target: 'price'. In order to keep this information in our model, we need to one hot encode this column.

In [30]:
```
encoder = OneHotEncoder(sparse=False, drop='first')
cat_cols=['zipcode']
data_ohe = encoder.fit_transform(df[cat_cols])
df_ohe = pd.DataFrame(data_ohe, columns=encoder.get_feature_names(cat_cols), index=df.i
df_ohe
```

Out[30]:

	zipcode_98002	zipcode_98003	zipcode_98004	zipcode_98005	zipcode_98006	zipcode_98007	z
0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	
21592	0.0	0.0	0.0	0.0	0.0	0.0	
21593	0.0	0.0	0.0	0.0	0.0	0.0	
21594	0.0	0.0	0.0	0.0	0.0	0.0	
21595	0.0	0.0	0.0	0.0	0.0	0.0	
21596	0.0	0.0	0.0	0.0	0.0	0.0	

21597 rows × 69 columns

In [31]:
```
df_ohe = pd.concat([df.drop('zipcode', axis=1), df_ohe], axis=1)
df_ohe
```

Out[31]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfron
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650	1.0	0.

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfron
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242	2.0	0.
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000	1.0	0.
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000	1.0	0.
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080	1.0	0.
...	...	...	...	...	...	...	...	...	.
21592	263000018	5/21/2014	360000.0	3	2.50	1530	1131	3.0	0.
21593	6600060120	2/23/2015	400000.0	4	2.50	2310	5813	2.0	0.
21594	1523300141	6/23/2014	402101.0	2	0.75	1020	1350	2.0	0.
21595	291310100	1/16/2015	400000.0	3	2.50	1600	2388	2.0	0.
21596	1523300157	10/15/2014	325000.0	2	0.75	1020	1076	2.0	0.

21597 rows × 91 columns

We are ready to fit the data with a linear regression model now. Since the building and improving on the model will be an iterative process, we can write a function that will not only create the model but also show us our QQ Plot as well as the residual information so we can check for normality and homoscedasticity.

Initial Model Prior to Addressing Multicollinearity

In [32]:
```
def model_lin_reg(df, target='price'):

    features = ' + '.join(df.drop(target, axis=1).columns)
    f  = f"{target}~"+features
    model = smf.ols(f, df).fit()
    display(model.summary())
    fig, ax = plt.subplots(ncols=2, figsize=(15,5))
    sm.graphics.qqplot(model.resid,line='45',fit=True, ax=ax[0])
    sns.scatterplot(x=model.predict(df, transform=True), y=model.resid, ax=ax[1])
    ax[1].set_ylabel('Residuals')
    ax[1].set_xlabel('Predicted')
    plt.axhline();
    return model
```

In [33]:
```
model_lin_reg(df=df_ohe)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.813
Model:	OLS	**Adj. R-squared:**	0.809
Method:	Least Squares	**F-statistic:**	201.2
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:12:51	**Log-Likelihood:**	-2.8926e+05
No. Observations:	21597	**AIC:**	5.794e+05

Df Residuals:	21138	**BIC:**	5.831e+05
Df Model:	458		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.545e+07	6.19e+06	-4.112	0.000	-3.76e+07	-1.33e+07
date[T.1/12/2015]	6.803e+04	1.62e+05	0.419	0.675	-2.5e+05	3.87e+05
date[T.1/13/2015]	7.407e+04	1.62e+05	0.456	0.648	-2.44e+05	3.92e+05
date[T.1/14/2015]	3.766e+04	1.62e+05	0.232	0.816	-2.8e+05	3.55e+05
date[T.1/15/2015]	5.788e+04	1.62e+05	0.357	0.721	-2.6e+05	3.76e+05
date[T.1/16/2015]	1.319e+04	1.62e+05	0.081	0.935	-3.04e+05	3.31e+05
date[T.1/17/2015]	1.532e+05	2.27e+05	0.675	0.500	-2.92e+05	5.98e+05
date[T.1/19/2015]	-1.373e+04	1.7e+05	-0.081	0.936	-3.48e+05	3.2e+05
date[T.1/2/2015]	7.584e+04	1.62e+05	0.467	0.640	-2.42e+05	3.94e+05
date[T.1/20/2015]	7.568e+04	1.62e+05	0.466	0.641	-2.42e+05	3.94e+05
date[T.1/21/2015]	7.04e+04	1.62e+05	0.435	0.664	-2.47e+05	3.88e+05
date[T.1/22/2015]	5.225e+04	1.62e+05	0.322	0.747	-2.66e+05	3.7e+05
date[T.1/23/2015]	3.699e+04	1.62e+05	0.228	0.820	-2.81e+05	3.55e+05
date[T.1/24/2015]	-1.045e+05	1.8e+05	-0.582	0.560	-4.56e+05	2.47e+05
date[T.1/25/2015]	3.207e+04	1.85e+05	0.173	0.863	-3.31e+05	3.95e+05
date[T.1/26/2015]	1.453e+04	1.62e+05	0.090	0.929	-3.03e+05	3.33e+05
date[T.1/27/2015]	7.388e+04	1.62e+05	0.456	0.648	-2.44e+05	3.91e+05
date[T.1/28/2015]	4.985e+04	1.62e+05	0.308	0.758	-2.67e+05	3.67e+05
date[T.1/29/2015]	7.743e+04	1.62e+05	0.477	0.634	-2.41e+05	3.96e+05
date[T.1/30/2015]	1.13e+05	1.63e+05	0.694	0.488	-2.06e+05	4.32e+05
date[T.1/31/2015]	-6.142e+04	2.27e+05	-0.270	0.787	-5.07e+05	3.84e+05
date[T.1/5/2015]	6.731e+04	1.62e+05	0.416	0.678	-2.5e+05	3.85e+05
date[T.1/6/2015]	7.594e+04	1.62e+05	0.467	0.640	-2.42e+05	3.94e+05
date[T.1/7/2015]	7.751e+04	1.62e+05	0.478	0.632	-2.4e+05	3.95e+05
date[T.1/8/2015]	3.653e+04	1.62e+05	0.225	0.822	-2.81e+05	3.54e+05
date[T.1/9/2015]	1.119e+05	1.63e+05	0.687	0.492	-2.07e+05	4.31e+05
date[T.10/1/2014]	6.697e+04	1.62e+05	0.415	0.678	-2.5e+05	3.84e+05
date[T.10/10/2014]	5.141e+04	1.62e+05	0.318	0.750	-2.65e+05	3.68e+05
date[T.10/11/2014]	6.385e+05	1.97e+05	3.240	0.001	2.52e+05	1.02e+06
date[T.10/12/2014]	7.394e+04	1.86e+05	0.399	0.690	-2.9e+05	4.38e+05

date[T.10/13/2014]	1.023e+05	1.62e+05	0.632	0.527	-2.15e+05	4.19e+05
date[T.10/14/2014]	5.965e+04	1.61e+05	0.369	0.712	-2.57e+05	3.76e+05
date[T.10/15/2014]	3.97e+04	1.61e+05	0.246	0.806	-2.77e+05	3.56e+05
date[T.10/16/2014]	4.875e+04	1.61e+05	0.302	0.763	-2.68e+05	3.65e+05
date[T.10/17/2014]	5.381e+04	1.62e+05	0.333	0.739	-2.63e+05	3.71e+05
date[T.10/18/2014]	1.243e+05	1.73e+05	0.716	0.474	-2.16e+05	4.64e+05
date[T.10/19/2014]	5.763e+04	1.8e+05	0.321	0.748	-2.94e+05	4.1e+05
date[T.10/2/2014]	4.299e+04	1.62e+05	0.266	0.790	-2.74e+05	3.6e+05
date[T.10/20/2014]	8.606e+04	1.62e+05	0.533	0.594	-2.31e+05	4.03e+05
date[T.10/21/2014]	3.387e+04	1.61e+05	0.210	0.834	-2.83e+05	3.5e+05
date[T.10/22/2014]	4.807e+04	1.62e+05	0.298	0.766	-2.69e+05	3.65e+05
date[T.10/23/2014]	3.916e+04	1.62e+05	0.242	0.809	-2.78e+05	3.56e+05
date[T.10/24/2014]	2.043e+04	1.62e+05	0.126	0.900	-2.97e+05	3.38e+05
date[T.10/25/2014]	1.032e+05	1.76e+05	0.587	0.557	-2.42e+05	4.48e+05
date[T.10/26/2014]	-5294.3178	1.8e+05	-0.029	0.976	-3.57e+05	3.47e+05
date[T.10/27/2014]	2.828e+04	1.61e+05	0.175	0.861	-2.88e+05	3.45e+05
date[T.10/28/2014]	4.598e+04	1.61e+05	0.285	0.776	-2.7e+05	3.62e+05
date[T.10/29/2014]	7.268e+04	1.61e+05	0.450	0.653	-2.44e+05	3.89e+05
date[T.10/3/2014]	4.948e+04	1.62e+05	0.306	0.760	-2.68e+05	3.67e+05
date[T.10/30/2014]	8.9e+04	1.62e+05	0.551	0.582	-2.28e+05	4.06e+05
date[T.10/31/2014]	2.422e+04	1.63e+05	0.149	0.882	-2.94e+05	3.43e+05
date[T.10/4/2014]	4334.6301	1.8e+05	0.024	0.981	-3.48e+05	3.56e+05
date[T.10/5/2014]	-2239.1184	1.85e+05	-0.012	0.990	-3.66e+05	3.61e+05
date[T.10/6/2014]	3.892e+04	1.62e+05	0.241	0.810	-2.78e+05	3.56e+05
date[T.10/7/2014]	5.128e+04	1.61e+05	0.318	0.751	-2.65e+05	3.68e+05
date[T.10/8/2014]	3.032e+04	1.62e+05	0.188	0.851	-2.87e+05	3.47e+05
date[T.10/9/2014]	4.504e+04	1.62e+05	0.279	0.780	-2.72e+05	3.62e+05
date[T.11/1/2014]	7.81e+04	1.76e+05	0.444	0.657	-2.67e+05	4.23e+05
date[T.11/10/2014]	8.014e+04	1.62e+05	0.496	0.620	-2.37e+05	3.97e+05
date[T.11/11/2014]	2.938e+04	1.62e+05	0.181	0.856	-2.89e+05	3.47e+05
date[T.11/12/2014]	2.512e+04	1.62e+05	0.155	0.876	-2.92e+05	3.42e+05
date[T.11/13/2014]	5.822e+04	1.61e+05	0.361	0.718	-2.58e+05	3.75e+05
date[T.11/14/2014]	6.781e+04	1.62e+05	0.419	0.675	-2.49e+05	3.85e+05
date[T.11/15/2014]	1.537e+05	1.97e+05	0.782	0.435	-2.32e+05	5.39e+05
date[T.11/16/2014]	-6662.0780	1.85e+05	-0.036	0.971	-3.7e+05	3.57e+05

date[T.11/17/2014]	3.116e+04	1.62e+05	0.193	0.847	-2.85e+05	3.48e+05
date[T.11/18/2014]	6.164e+04	1.61e+05	0.382	0.703	-2.55e+05	3.78e+05
date[T.11/19/2014]	8.054e+04	1.62e+05	0.499	0.618	-2.36e+05	3.97e+05
date[T.11/2/2014]	1.222e+05	2.27e+05	0.538	0.591	-3.23e+05	5.67e+05
date[T.11/20/2014]	5.771e+04	1.62e+05	0.357	0.721	-2.59e+05	3.74e+05
date[T.11/21/2014]	2.835e+04	1.62e+05	0.176	0.861	-2.88e+05	3.45e+05
date[T.11/22/2014]	7.583e+04	1.72e+05	0.442	0.659	-2.61e+05	4.12e+05
date[T.11/23/2014]	1.032e+05	1.8e+05	0.575	0.565	-2.49e+05	4.55e+05
date[T.11/24/2014]	7.264e+04	1.62e+05	0.450	0.653	-2.44e+05	3.89e+05
date[T.11/25/2014]	5.312e+04	1.62e+05	0.329	0.743	-2.64e+05	3.7e+05
date[T.11/26/2014]	5.529e+04	1.62e+05	0.341	0.733	-2.63e+05	3.73e+05
date[T.11/28/2014]	6.328e+04	1.97e+05	0.322	0.748	-3.22e+05	4.49e+05
date[T.11/29/2014]	7.361e+04	1.85e+05	0.397	0.691	-2.9e+05	4.37e+05
date[T.11/3/2014]	4.359e+04	1.62e+05	0.270	0.787	-2.73e+05	3.6e+05
date[T.11/30/2014]	2.687e+05	2.27e+05	1.183	0.237	-1.77e+05	7.14e+05
date[T.11/4/2014]	4.77e+04	1.62e+05	0.295	0.768	-2.69e+05	3.65e+05
date[T.11/5/2014]	4.512e+04	1.62e+05	0.279	0.780	-2.72e+05	3.62e+05
date[T.11/6/2014]	6.82e+04	1.62e+05	0.422	0.673	-2.49e+05	3.85e+05
date[T.11/7/2014]	3.051e+04	1.62e+05	0.189	0.850	-2.86e+05	3.47e+05
date[T.11/8/2014]	1.371e+05	1.76e+05	0.780	0.436	-2.08e+05	4.82e+05
date[T.11/9/2014]	-1.242e+04	1.85e+05	-0.067	0.947	-3.76e+05	3.51e+05
date[T.12/1/2014]	2.633e+04	1.61e+05	0.163	0.870	-2.9e+05	3.43e+05
date[T.12/10/2014]	5.943e+04	1.62e+05	0.368	0.713	-2.57e+05	3.76e+05
date[T.12/11/2014]	4.551e+04	1.62e+05	0.282	0.778	-2.71e+05	3.62e+05
date[T.12/12/2014]	6.022e+04	1.62e+05	0.372	0.710	-2.57e+05	3.77e+05
date[T.12/13/2014]	3.757e+04	1.8e+05	0.209	0.834	-3.14e+05	3.9e+05
date[T.12/14/2014]	8.817e+04	1.8e+05	0.491	0.623	-2.64e+05	4.4e+05
date[T.12/15/2014]	3.594e+04	1.62e+05	0.222	0.824	-2.81e+05	3.53e+05
date[T.12/16/2014]	5.408e+04	1.62e+05	0.334	0.738	-2.63e+05	3.71e+05
date[T.12/17/2014]	5.331e+04	1.62e+05	0.329	0.742	-2.64e+05	3.7e+05
date[T.12/18/2014]	4.639e+04	1.62e+05	0.287	0.774	-2.71e+05	3.63e+05
date[T.12/19/2014]	1.028e+05	1.62e+05	0.633	0.527	-2.16e+05	4.21e+05
date[T.12/2/2014]	2.78e+04	1.61e+05	0.172	0.863	-2.89e+05	3.44e+05
date[T.12/20/2014]	1.079e+04	1.7e+05	0.063	0.950	-3.23e+05	3.45e+05
date[T.12/21/2014]	5.639e+04	1.97e+05	0.287	0.774	-3.29e+05	4.42e+05

date[T.12/22/2014]	4.044e+04	1.62e+05	0.250	0.803	-2.77e+05	3.58e+05
date[T.12/23/2014]	5.606e+04	1.62e+05	0.346	0.729	-2.61e+05	3.73e+05
date[T.12/24/2014]	7.062e+04	1.63e+05	0.433	0.665	-2.49e+05	3.9e+05
date[T.12/26/2014]	4.028e+04	1.63e+05	0.248	0.804	-2.79e+05	3.59e+05
date[T.12/27/2014]	2.754e+04	1.97e+05	0.140	0.889	-3.58e+05	4.13e+05
date[T.12/29/2014]	5.676e+04	1.62e+05	0.351	0.726	-2.6e+05	3.74e+05
date[T.12/3/2014]	6.593e+04	1.62e+05	0.408	0.683	-2.51e+05	3.83e+05
date[T.12/30/2014]	1.047e+05	1.62e+05	0.645	0.519	-2.14e+05	4.23e+05
date[T.12/31/2014]	7.635e+04	1.62e+05	0.470	0.638	-2.42e+05	3.95e+05
date[T.12/4/2014]	4.659e+04	1.62e+05	0.288	0.773	-2.7e+05	3.64e+05
date[T.12/5/2014]	8.073e+04	1.62e+05	0.499	0.618	-2.37e+05	3.98e+05
date[T.12/6/2014]	4.309e+04	1.73e+05	0.248	0.804	-2.97e+05	3.83e+05
date[T.12/7/2014]	-2.062e+04	1.97e+05	-0.105	0.916	-4.06e+05	3.65e+05
date[T.12/8/2014]	6.5e+04	1.62e+05	0.402	0.688	-2.52e+05	3.82e+05
date[T.12/9/2014]	6.11e+04	1.62e+05	0.378	0.705	-2.56e+05	3.78e+05
date[T.2/1/2015]	1.146e+05	1.97e+05	0.583	0.560	-2.71e+05	5e+05
date[T.2/10/2015]	4.414e+04	1.62e+05	0.272	0.785	-2.73e+05	3.62e+05
date[T.2/11/2015]	7.121e+04	1.62e+05	0.440	0.660	-2.46e+05	3.88e+05
date[T.2/12/2015]	7.664e+04	1.62e+05	0.472	0.637	-2.41e+05	3.95e+05
date[T.2/13/2015]	8.436e+04	1.62e+05	0.522	0.602	-2.33e+05	4.01e+05
date[T.2/14/2015]	2.728e+04	1.85e+05	0.147	0.883	-3.36e+05	3.91e+05
date[T.2/15/2015]	9.796e+04	2.27e+05	0.431	0.666	-3.47e+05	5.43e+05
date[T.2/16/2015]	5.582e+04	1.73e+05	0.322	0.748	-2.84e+05	3.96e+05
date[T.2/17/2015]	5.242e+04	1.62e+05	0.324	0.746	-2.65e+05	3.69e+05
date[T.2/18/2015]	5.559e+04	1.61e+05	0.344	0.731	-2.61e+05	3.72e+05
date[T.2/19/2015]	7.238e+04	1.62e+05	0.448	0.654	-2.44e+05	3.89e+05
date[T.2/2/2015]	7.324e+04	1.62e+05	0.451	0.652	-2.45e+05	3.91e+05
date[T.2/20/2015]	7.888e+04	1.62e+05	0.488	0.626	-2.38e+05	3.96e+05
date[T.2/21/2015]	1.024e+05	1.85e+05	0.552	0.581	-2.61e+05	4.66e+05
date[T.2/22/2015]	2.317e+04	1.67e+05	0.139	0.889	-3.03e+05	3.5e+05
date[T.2/23/2015]	5.887e+04	1.62e+05	0.364	0.716	-2.58e+05	3.76e+05
date[T.2/24/2015]	5.066e+04	1.61e+05	0.314	0.754	-2.66e+05	3.67e+05
date[T.2/25/2015]	9.108e+04	1.61e+05	0.564	0.573	-2.25e+05	4.07e+05
date[T.2/26/2015]	5.819e+04	1.62e+05	0.359	0.719	-2.59e+05	3.76e+05
date[T.2/27/2015]	8.307e+04	1.62e+05	0.512	0.608	-2.35e+05	4.01e+05

date[T.2/28/2015]	1.327e+05	1.76e+05	0.755	0.451	-2.12e+05	4.78e+05
date[T.2/3/2015]	3.639e+04	1.62e+05	0.224	0.823	-2.82e+05	3.54e+05
date[T.2/4/2015]	7.95e+04	1.62e+05	0.491	0.624	-2.38e+05	3.97e+05
date[T.2/5/2015]	4.569e+04	1.62e+05	0.282	0.778	-2.72e+05	3.64e+05
date[T.2/6/2015]	7.031e+04	1.62e+05	0.434	0.665	-2.47e+05	3.88e+05
date[T.2/7/2015]	4.109e+04	1.85e+05	0.222	0.825	-3.22e+05	4.05e+05
date[T.2/9/2015]	7.098e+04	1.62e+05	0.438	0.661	-2.47e+05	3.89e+05
date[T.3/1/2015]	2.909e+04	1.73e+05	0.168	0.867	-3.11e+05	3.69e+05
date[T.3/10/2015]	6.837e+04	1.62e+05	0.423	0.672	-2.48e+05	3.85e+05
date[T.3/11/2015]	7.258e+04	1.61e+05	0.449	0.653	-2.44e+05	3.89e+05
date[T.3/12/2015]	7.413e+04	1.62e+05	0.459	0.647	-2.43e+05	3.91e+05
date[T.3/13/2015]	1.209e+05	1.62e+05	0.748	0.455	-1.96e+05	4.38e+05
date[T.3/14/2015]	1.021e+05	1.73e+05	0.588	0.556	-2.38e+05	4.42e+05
date[T.3/15/2015]	9.327e+04	1.85e+05	0.503	0.615	-2.7e+05	4.57e+05
date[T.3/16/2015]	1.093e+05	1.62e+05	0.677	0.498	-2.07e+05	4.26e+05
date[T.3/17/2015]	8.078e+04	1.61e+05	0.500	0.617	-2.36e+05	3.97e+05
date[T.3/18/2015]	7.683e+04	1.61e+05	0.476	0.634	-2.4e+05	3.93e+05
date[T.3/19/2015]	9.148e+04	1.62e+05	0.566	0.572	-2.25e+05	4.08e+05
date[T.3/2/2015]	1.286e+05	1.62e+05	0.791	0.429	-1.9e+05	4.47e+05
date[T.3/20/2015]	6.824e+04	1.62e+05	0.422	0.673	-2.49e+05	3.85e+05
date[T.3/21/2015]	6.505e+04	1.66e+05	0.392	0.695	-2.6e+05	3.9e+05
date[T.3/22/2015]	2.261e+05	1.8e+05	1.259	0.208	-1.26e+05	5.78e+05
date[T.3/23/2015]	8.544e+04	1.61e+05	0.529	0.597	-2.31e+05	4.02e+05
date[T.3/24/2015]	7.513e+04	1.61e+05	0.466	0.641	-2.41e+05	3.91e+05
date[T.3/25/2015]	8.621e+04	1.61e+05	0.535	0.593	-2.3e+05	4.02e+05
date[T.3/26/2015]	1.082e+05	1.61e+05	0.671	0.502	-2.08e+05	4.24e+05
date[T.3/27/2015]	7.932e+04	1.61e+05	0.492	0.623	-2.37e+05	3.95e+05
date[T.3/28/2015]	7.601e+04	1.7e+05	0.446	0.655	-2.58e+05	4.1e+05
date[T.3/29/2015]	-1.98e+04	1.7e+05	-0.116	0.907	-3.54e+05	3.14e+05
date[T.3/3/2015]	1.295e+05	1.62e+05	0.800	0.424	-1.88e+05	4.47e+05
date[T.3/30/2015]	7.789e+04	1.61e+05	0.482	0.630	-2.39e+05	3.94e+05
date[T.3/31/2015]	9.818e+04	1.62e+05	0.607	0.544	-2.19e+05	4.15e+05
date[T.3/4/2015]	9.591e+04	1.61e+05	0.594	0.552	-2.2e+05	4.12e+05
date[T.3/5/2015]	7.057e+04	1.62e+05	0.437	0.662	-2.46e+05	3.87e+05
date[T.3/6/2015]	8.419e+04	1.62e+05	0.520	0.603	-2.33e+05	4.02e+05

date[T.3/7/2015]	8.48e+04	1.79e+05	0.473	0.637	-2.67e+05	4.37e+05
date[T.3/8/2015]	1.941e+05	2.27e+05	0.854	0.393	-2.51e+05	6.39e+05
date[T.3/9/2015]	7.361e+04	1.62e+05	0.455	0.649	-2.43e+05	3.91e+05
date[T.4/1/2015]	1.051e+05	1.61e+05	0.651	0.515	-2.11e+05	4.22e+05
date[T.4/10/2015]	9.357e+04	1.62e+05	0.579	0.563	-2.23e+05	4.1e+05
date[T.4/11/2015]	7.553e+04	1.67e+05	0.453	0.650	-2.51e+05	4.02e+05
date[T.4/12/2015]	5.507e+04	1.67e+05	0.330	0.741	-2.72e+05	3.82e+05
date[T.4/13/2015]	1.319e+05	1.61e+05	0.817	0.414	-1.85e+05	4.48e+05
date[T.4/14/2015]	7.981e+04	1.61e+05	0.495	0.621	-2.36e+05	3.96e+05
date[T.4/15/2015]	1.128e+05	1.62e+05	0.698	0.485	-2.04e+05	4.3e+05
date[T.4/16/2015]	7.432e+04	1.62e+05	0.460	0.646	-2.42e+05	3.91e+05
date[T.4/17/2015]	7.595e+04	1.62e+05	0.470	0.638	-2.41e+05	3.93e+05
date[T.4/18/2015]	1.07e+05	1.76e+05	0.608	0.543	-2.38e+05	4.52e+05
date[T.4/19/2015]	9.229e+04	1.73e+05	0.532	0.595	-2.48e+05	4.32e+05
date[T.4/2/2015]	9.088e+04	1.61e+05	0.563	0.573	-2.25e+05	4.07e+05
date[T.4/20/2015]	5.602e+04	1.62e+05	0.347	0.729	-2.61e+05	3.73e+05
date[T.4/21/2015]	1.008e+05	1.61e+05	0.625	0.532	-2.15e+05	4.17e+05
date[T.4/22/2015]	8.13e+04	1.61e+05	0.504	0.614	-2.35e+05	3.97e+05
date[T.4/23/2015]	8.358e+04	1.61e+05	0.518	0.604	-2.33e+05	4e+05
date[T.4/24/2015]	1.005e+05	1.61e+05	0.623	0.533	-2.16e+05	4.17e+05
date[T.4/25/2015]	1.168e+05	1.66e+05	0.706	0.480	-2.08e+05	4.41e+05
date[T.4/26/2015]	7.091e+04	1.67e+05	0.426	0.670	-2.56e+05	3.98e+05
date[T.4/27/2015]	7.666e+04	1.61e+05	0.476	0.634	-2.39e+05	3.93e+05
date[T.4/28/2015]	9.376e+04	1.61e+05	0.582	0.561	-2.22e+05	4.1e+05
date[T.4/29/2015]	1.136e+05	1.61e+05	0.704	0.481	-2.02e+05	4.3e+05
date[T.4/3/2015]	1.089e+05	1.62e+05	0.674	0.500	-2.08e+05	4.26e+05
date[T.4/30/2015]	9.605e+04	1.62e+05	0.595	0.552	-2.21e+05	4.13e+05
date[T.4/4/2015]	1.215e+05	1.8e+05	0.677	0.498	-2.3e+05	4.73e+05
date[T.4/5/2015]	8.902e+04	1.73e+05	0.513	0.608	-2.51e+05	4.29e+05
date[T.4/6/2015]	7.828e+04	1.62e+05	0.485	0.628	-2.38e+05	3.95e+05
date[T.4/7/2015]	1.232e+05	1.61e+05	0.764	0.445	-1.93e+05	4.39e+05
date[T.4/8/2015]	8.854e+04	1.61e+05	0.549	0.583	-2.28e+05	4.05e+05
date[T.4/9/2015]	7.992e+04	1.61e+05	0.495	0.621	-2.37e+05	3.96e+05
date[T.5/1/2015]	1.236e+05	1.62e+05	0.765	0.444	-1.93e+05	4.4e+05
date[T.5/10/2014]	7.679e+04	1.76e+05	0.437	0.662	-2.68e+05	4.22e+05

date[T.5/10/2015]	5.498e+04	1.97e+05	0.280	0.780	-3.31e+05	4.41e+05
date[T.5/11/2014]	9.365e+04	1.97e+05	0.476	0.634	-2.92e+05	4.79e+05
date[T.5/11/2015]	1.335e+05	1.63e+05	0.821	0.412	-1.85e+05	4.52e+05
date[T.5/12/2014]	1.383e+04	1.62e+05	0.086	0.932	-3.03e+05	3.31e+05
date[T.5/12/2015]	1.037e+05	1.62e+05	0.639	0.523	-2.14e+05	4.22e+05
date[T.5/13/2014]	6.619e+04	1.61e+05	0.410	0.682	-2.5e+05	3.83e+05
date[T.5/13/2015]	1.146e+05	1.63e+05	0.703	0.482	-2.05e+05	4.34e+05
date[T.5/14/2014]	4.359e+04	1.62e+05	0.270	0.787	-2.73e+05	3.6e+05
date[T.5/14/2015]	1.908e+05	1.68e+05	1.138	0.255	-1.38e+05	5.19e+05
date[T.5/15/2014]	3.854e+04	1.62e+05	0.239	0.811	-2.78e+05	3.55e+05
date[T.5/15/2015]	5.578e+04	2.27e+05	0.245	0.806	-3.9e+05	5.01e+05
date[T.5/16/2014]	4.014e+04	1.62e+05	0.248	0.804	-2.77e+05	3.57e+05
date[T.5/17/2014]	-6.369e+04	2.27e+05	-0.280	0.779	-5.09e+05	3.82e+05
date[T.5/18/2014]	7.263e+04	1.72e+05	0.423	0.672	-2.64e+05	4.09e+05
date[T.5/19/2014]	5.975e+04	1.62e+05	0.370	0.711	-2.57e+05	3.76e+05
date[T.5/2/2014]	4.136e+04	1.62e+05	0.256	0.798	-2.76e+05	3.58e+05
date[T.5/2/2015]	1.002e+05	1.73e+05	0.578	0.564	-2.4e+05	4.4e+05
date[T.5/20/2014]	4.395e+04	1.61e+05	0.273	0.785	-2.72e+05	3.6e+05
date[T.5/21/2014]	4.668e+04	1.61e+05	0.289	0.772	-2.7e+05	3.63e+05
date[T.5/22/2014]	3.296e+04	1.61e+05	0.204	0.838	-2.83e+05	3.49e+05
date[T.5/23/2014]	2.925e+04	1.62e+05	0.181	0.856	-2.87e+05	3.46e+05
date[T.5/24/2014]	8.123e+04	1.68e+05	0.484	0.628	-2.48e+05	4.1e+05
date[T.5/24/2015]	6.2e+04	2.27e+05	0.273	0.785	-3.83e+05	5.07e+05
date[T.5/25/2014]	-3.29e+04	1.76e+05	-0.187	0.852	-3.78e+05	3.12e+05
date[T.5/26/2014]	9.481e+04	1.7e+05	0.557	0.578	-2.39e+05	4.29e+05
date[T.5/27/2014]	4.602e+04	1.61e+05	0.285	0.775	-2.7e+05	3.62e+05
date[T.5/27/2015]	3.727e+05	2.27e+05	1.641	0.101	-7.24e+04	8.18e+05
date[T.5/28/2014]	4.93e+04	1.61e+05	0.306	0.760	-2.67e+05	3.65e+05
date[T.5/29/2014]	7.221e+04	1.62e+05	0.447	0.655	-2.45e+05	3.89e+05
date[T.5/3/2014]	1.532e+05	1.8e+05	0.853	0.394	-1.99e+05	5.05e+05
date[T.5/3/2015]	1.184e+05	1.68e+05	0.703	0.482	-2.12e+05	4.49e+05
date[T.5/30/2014]	5.897e+04	1.62e+05	0.364	0.716	-2.58e+05	3.76e+05
date[T.5/31/2014]	5.279e+04	1.73e+05	0.304	0.761	-2.87e+05	3.93e+05
date[T.5/4/2014]	-9806.1064	1.76e+05	-0.056	0.956	-3.55e+05	3.35e+05
date[T.5/4/2015]	8.65e+04	1.61e+05	0.536	0.592	-2.3e+05	4.03e+05

date[T.5/5/2014]	3.339e+04	1.62e+05	0.207	0.836	-2.83e+05	3.5e+05
date[T.5/5/2015]	1.1e+05	1.61e+05	0.681	0.496	-2.06e+05	4.26e+05
date[T.5/6/2014]	3.796e+04	1.62e+05	0.235	0.814	-2.79e+05	3.55e+05
date[T.5/6/2015]	1.098e+05	1.61e+05	0.680	0.497	-2.07e+05	4.26e+05
date[T.5/7/2014]	5.086e+04	1.61e+05	0.315	0.753	-2.66e+05	3.67e+05
date[T.5/7/2015]	9.735e+04	1.62e+05	0.602	0.547	-2.19e+05	4.14e+05
date[T.5/8/2014]	5.61e+04	1.62e+05	0.347	0.728	-2.61e+05	3.73e+05
date[T.5/8/2015]	1.121e+05	1.62e+05	0.692	0.489	-2.05e+05	4.3e+05
date[T.5/9/2014]	4.127e+04	1.62e+05	0.255	0.798	-2.75e+05	3.58e+05
date[T.5/9/2015]	5.752e+04	1.85e+05	0.310	0.756	-3.06e+05	4.21e+05
date[T.6/1/2014]	1.317e+05	1.72e+05	0.767	0.443	-2.05e+05	4.68e+05
date[T.6/10/2014]	5.979e+04	1.61e+05	0.370	0.711	-2.57e+05	3.76e+05
date[T.6/11/2014]	8.144e+04	1.61e+05	0.504	0.614	-2.35e+05	3.98e+05
date[T.6/12/2014]	5.128e+04	1.61e+05	0.318	0.751	-2.65e+05	3.68e+05
date[T.6/13/2014]	7.397e+04	1.62e+05	0.458	0.647	-2.43e+05	3.91e+05
date[T.6/14/2014]	8.621e+04	1.7e+05	0.506	0.613	-2.48e+05	4.2e+05
date[T.6/15/2014]	-2.909e+04	1.72e+05	-0.169	0.865	-3.66e+05	3.07e+05
date[T.6/16/2014]	5.303e+04	1.61e+05	0.329	0.742	-2.63e+05	3.69e+05
date[T.6/17/2014]	8.244e+04	1.61e+05	0.511	0.609	-2.34e+05	3.99e+05
date[T.6/18/2014]	8.762e+04	1.61e+05	0.543	0.587	-2.29e+05	4.04e+05
date[T.6/19/2014]	4.913e+04	1.61e+05	0.304	0.761	-2.67e+05	3.65e+05
date[T.6/2/2014]	4.202e+04	1.62e+05	0.260	0.795	-2.75e+05	3.59e+05
date[T.6/20/2014]	6.359e+04	1.61e+05	0.394	0.693	-2.53e+05	3.8e+05
date[T.6/21/2014]	4.655e+04	1.7e+05	0.273	0.785	-2.87e+05	3.8e+05
date[T.6/22/2014]	3.956e+04	1.67e+05	0.237	0.813	-2.88e+05	3.67e+05
date[T.6/23/2014]	5.032e+04	1.61e+05	0.312	0.755	-2.65e+05	3.66e+05
date[T.6/24/2014]	6.552e+04	1.61e+05	0.406	0.684	-2.51e+05	3.82e+05
date[T.6/25/2014]	6.899e+04	1.61e+05	0.428	0.669	-2.47e+05	3.85e+05
date[T.6/26/2014]	4.393e+04	1.61e+05	0.273	0.785	-2.72e+05	3.6e+05
date[T.6/27/2014]	2.525e+04	1.61e+05	0.156	0.876	-2.91e+05	3.42e+05
date[T.6/28/2014]	9.41e+04	1.67e+05	0.565	0.572	-2.33e+05	4.21e+05
date[T.6/29/2014]	-2.383e+04	1.72e+05	-0.139	0.890	-3.6e+05	3.13e+05
date[T.6/3/2014]	5.187e+04	1.61e+05	0.322	0.748	-2.64e+05	3.68e+05
date[T.6/30/2014]	4.908e+04	1.62e+05	0.304	0.761	-2.68e+05	3.66e+05
date[T.6/4/2014]	8.561e+04	1.61e+05	0.531	0.596	-2.31e+05	4.02e+05

date[T.6/5/2014]	3.913e+04	1.61e+05	0.242	0.808	-2.77e+05	3.56e+05
date[T.6/6/2014]	5.061e+04	1.62e+05	0.313	0.754	-2.66e+05	3.68e+05
date[T.6/7/2014]	-3.891e+04	1.8e+05	-0.217	0.828	-3.91e+05	3.13e+05
date[T.6/8/2014]	9.254e+04	1.68e+05	0.549	0.583	-2.38e+05	4.23e+05
date[T.6/9/2014]	4.664e+04	1.61e+05	0.289	0.773	-2.7e+05	3.63e+05
date[T.7/1/2014]	7.264e+04	1.61e+05	0.450	0.652	-2.43e+05	3.89e+05
date[T.7/10/2014]	4.922e+04	1.61e+05	0.305	0.760	-2.67e+05	3.66e+05
date[T.7/11/2014]	6.856e+04	1.62e+05	0.424	0.671	-2.48e+05	3.85e+05
date[T.7/12/2014]	6.368e+04	1.69e+05	0.376	0.707	-2.68e+05	3.95e+05
date[T.7/13/2014]	5.458e+04	1.85e+05	0.294	0.768	-3.09e+05	4.18e+05
date[T.7/14/2014]	7.302e+04	1.61e+05	0.453	0.651	-2.43e+05	3.89e+05
date[T.7/15/2014]	4.392e+04	1.61e+05	0.272	0.785	-2.72e+05	3.6e+05
date[T.7/16/2014]	5.91e+04	1.61e+05	0.366	0.714	-2.57e+05	3.75e+05
date[T.7/17/2014]	4.475e+04	1.62e+05	0.277	0.782	-2.72e+05	3.61e+05
date[T.7/18/2014]	5.829e+04	1.61e+05	0.361	0.718	-2.58e+05	3.75e+05
date[T.7/19/2014]	9.856e+04	1.76e+05	0.560	0.575	-2.46e+05	4.43e+05
date[T.7/2/2014]	7.063e+04	1.61e+05	0.437	0.662	-2.46e+05	3.87e+05
date[T.7/20/2014]	-5623.3992	1.7e+05	-0.033	0.974	-3.4e+05	3.28e+05
date[T.7/21/2014]	4.591e+04	1.61e+05	0.284	0.776	-2.7e+05	3.62e+05
date[T.7/22/2014]	3.277e+04	1.61e+05	0.203	0.839	-2.83e+05	3.49e+05
date[T.7/23/2014]	4.054e+04	1.61e+05	0.251	0.802	-2.76e+05	3.57e+05
date[T.7/24/2014]	4.783e+04	1.61e+05	0.296	0.767	-2.69e+05	3.64e+05
date[T.7/25/2014]	6.36e+04	1.61e+05	0.394	0.693	-2.53e+05	3.8e+05
date[T.7/26/2014]	9.555e+04	1.7e+05	0.561	0.575	-2.38e+05	4.29e+05
date[T.7/27/2014]	1.243e+05	2.27e+05	0.547	0.584	-3.21e+05	5.69e+05
date[T.7/28/2014]	3.098e+04	1.61e+05	0.192	0.848	-2.85e+05	3.47e+05
date[T.7/29/2014]	3.81e+04	1.61e+05	0.236	0.813	-2.78e+05	3.54e+05
date[T.7/3/2014]	3.972e+04	1.62e+05	0.246	0.806	-2.77e+05	3.57e+05
date[T.7/30/2014]	4.537e+04	1.62e+05	0.281	0.779	-2.71e+05	3.62e+05
date[T.7/31/2014]	5.867e+04	1.62e+05	0.363	0.716	-2.58e+05	3.75e+05
date[T.7/4/2014]	9.084e+04	1.97e+05	0.462	0.644	-2.95e+05	4.76e+05
date[T.7/5/2014]	1.891e+05	1.73e+05	1.090	0.276	-1.51e+05	5.29e+05
date[T.7/6/2014]	7.43e+04	1.85e+05	0.401	0.689	-2.89e+05	4.38e+05
date[T.7/7/2014]	5.491e+04	1.62e+05	0.340	0.734	-2.62e+05	3.72e+05
date[T.7/8/2014]	3.754e+04	1.61e+05	0.233	0.816	-2.78e+05	3.54e+05

date[T.7/9/2014]	5.091e+04	1.61e+05	0.316	0.752	-2.65e+05	3.67e+05
date[T.8/1/2014]	5.964e+04	1.62e+05	0.369	0.712	-2.57e+05	3.76e+05
date[T.8/10/2014]	4.276e+04	1.85e+05	0.231	0.818	-3.21e+05	4.06e+05
date[T.8/11/2014]	6.691e+04	1.61e+05	0.415	0.678	-2.49e+05	3.83e+05
date[T.8/12/2014]	6.879e+04	1.61e+05	0.426	0.670	-2.47e+05	3.85e+05
date[T.8/13/2014]	4.686e+04	1.61e+05	0.290	0.772	-2.69e+05	3.63e+05
date[T.8/14/2014]	4.269e+04	1.61e+05	0.264	0.791	-2.74e+05	3.59e+05
date[T.8/15/2014]	8.373e+04	1.62e+05	0.517	0.605	-2.34e+05	4.01e+05
date[T.8/16/2014]	7.211e+04	1.8e+05	0.402	0.688	-2.8e+05	4.24e+05
date[T.8/17/2014]	2.482e+04	1.8e+05	0.138	0.890	-3.27e+05	3.77e+05
date[T.8/18/2014]	6.429e+04	1.62e+05	0.398	0.691	-2.52e+05	3.81e+05
date[T.8/19/2014]	7.431e+04	1.61e+05	0.460	0.645	-2.42e+05	3.91e+05
date[T.8/2/2014]	8.373e+04	1.76e+05	0.476	0.634	-2.61e+05	4.29e+05
date[T.8/20/2014]	5.201e+04	1.61e+05	0.323	0.747	-2.64e+05	3.68e+05
date[T.8/21/2014]	3.99e+04	1.61e+05	0.247	0.805	-2.77e+05	3.56e+05
date[T.8/22/2014]	3.269e+04	1.61e+05	0.203	0.839	-2.84e+05	3.49e+05
date[T.8/23/2014]	7.772e+04	1.72e+05	0.453	0.651	-2.59e+05	4.14e+05
date[T.8/24/2014]	5.157e+04	1.97e+05	0.262	0.793	-3.34e+05	4.37e+05
date[T.8/25/2014]	5.985e+04	1.61e+05	0.371	0.711	-2.56e+05	3.76e+05
date[T.8/26/2014]	5.291e+04	1.61e+05	0.328	0.743	-2.63e+05	3.69e+05
date[T.8/27/2014]	3.934e+04	1.61e+05	0.244	0.807	-2.77e+05	3.56e+05
date[T.8/28/2014]	6.143e+04	1.62e+05	0.380	0.704	-2.55e+05	3.78e+05
date[T.8/29/2014]	4.822e+04	1.62e+05	0.298	0.766	-2.69e+05	3.66e+05
date[T.8/3/2014]	8.831e+04	2.27e+05	0.389	0.697	-3.57e+05	5.33e+05
date[T.8/30/2014]	1.118e+05	2.28e+05	0.491	0.623	-3.34e+05	5.58e+05
date[T.8/31/2014]	7.317e+04	1.85e+05	0.395	0.693	-2.9e+05	4.37e+05
date[T.8/4/2014]	6.574e+04	1.61e+05	0.407	0.684	-2.51e+05	3.82e+05
date[T.8/5/2014]	5.734e+04	1.61e+05	0.355	0.722	-2.59e+05	3.74e+05
date[T.8/6/2014]	4.191e+04	1.62e+05	0.260	0.795	-2.75e+05	3.58e+05
date[T.8/7/2014]	6.086e+04	1.62e+05	0.376	0.707	-2.56e+05	3.78e+05
date[T.8/8/2014]	6.74e+04	1.62e+05	0.417	0.677	-2.49e+05	3.84e+05
date[T.8/9/2014]	1.033e+05	1.97e+05	0.525	0.600	-2.82e+05	4.89e+05
date[T.9/1/2014]	9.074e+04	1.73e+05	0.523	0.601	-2.49e+05	4.31e+05
date[T.9/10/2014]	4.157e+04	1.61e+05	0.257	0.797	-2.75e+05	3.58e+05
date[T.9/11/2014]	7.055e+04	1.61e+05	0.437	0.662	-2.46e+05	3.87e+05

date[T.9/12/2014]	1.498e+04	1.62e+05	0.093	0.926	-3.02e+05	3.32e+05
date[T.9/13/2014]	8.144e+04	1.76e+05	0.463	0.643	-2.63e+05	4.26e+05
date[T.9/14/2014]	6.964e+04	1.8e+05	0.388	0.698	-2.82e+05	4.22e+05
date[T.9/15/2014]	2.052e+04	1.62e+05	0.127	0.899	-2.96e+05	3.37e+05
date[T.9/16/2014]	3.815e+04	1.61e+05	0.236	0.813	-2.78e+05	3.55e+05
date[T.9/17/2014]	2.567e+04	1.62e+05	0.159	0.874	-2.91e+05	3.43e+05
date[T.9/18/2014]	4.86e+04	1.62e+05	0.301	0.764	-2.68e+05	3.65e+05
date[T.9/19/2014]	7.247e+04	1.62e+05	0.448	0.654	-2.44e+05	3.89e+05
date[T.9/2/2014]	6.291e+04	1.62e+05	0.389	0.697	-2.54e+05	3.8e+05
date[T.9/20/2014]	9.853e+04	1.76e+05	0.560	0.575	-2.46e+05	4.43e+05
date[T.9/21/2014]	-2.162e+04	1.76e+05	-0.123	0.902	-3.66e+05	3.23e+05
date[T.9/22/2014]	5.286e+04	1.61e+05	0.327	0.743	-2.64e+05	3.69e+05
date[T.9/23/2014]	7.304e+04	1.61e+05	0.453	0.651	-2.43e+05	3.89e+05
date[T.9/24/2014]	2.913e+04	1.61e+05	0.180	0.857	-2.87e+05	3.45e+05
date[T.9/25/2014]	4.459e+04	1.62e+05	0.276	0.783	-2.72e+05	3.61e+05
date[T.9/26/2014]	4.775e+04	1.61e+05	0.296	0.767	-2.69e+05	3.64e+05
date[T.9/27/2014]	-1.319e+05	1.76e+05	-0.750	0.454	-4.77e+05	2.13e+05
date[T.9/28/2014]	8.464e+04	1.85e+05	0.456	0.648	-2.79e+05	4.48e+05
date[T.9/29/2014]	5.723e+04	1.62e+05	0.354	0.723	-2.59e+05	3.74e+05
date[T.9/3/2014]	6.344e+04	1.62e+05	0.393	0.694	-2.53e+05	3.8e+05
date[T.9/30/2014]	3.846e+04	1.62e+05	0.237	0.812	-2.79e+05	3.56e+05
date[T.9/4/2014]	5.176e+04	1.62e+05	0.320	0.749	-2.65e+05	3.68e+05
date[T.9/5/2014]	5.859e+04	1.61e+05	0.363	0.717	-2.58e+05	3.75e+05
date[T.9/6/2014]	3.968e+04	1.76e+05	0.226	0.822	-3.05e+05	3.85e+05
date[T.9/7/2014]	2.516e+04	1.97e+05	0.128	0.898	-3.6e+05	4.11e+05
date[T.9/8/2014]	6.088e+04	1.62e+05	0.377	0.706	-2.56e+05	3.78e+05
date[T.9/9/2014]	5.418e+04	1.61e+05	0.336	0.737	-2.62e+05	3.71e+05
id	-8.566e-07	3.99e-07	-2.148	0.032	-1.64e-06	-7.49e-08
bedrooms	-2.758e+04	1548.715	-17.811	0.000	-3.06e+04	-2.45e+04
bathrooms	2.604e+04	2671.682	9.747	0.000	2.08e+04	3.13e+04
sqft_living	96.8500	14.574	6.646	0.000	68.284	125.416
sqft_lot	0.2472	0.039	6.364	0.000	0.171	0.323
floors	-4.503e+04	3194.878	-14.095	0.000	-5.13e+04	-3.88e+04
waterfront	6.891e+05	1.48e+04	46.561	0.000	6.6e+05	7.18e+05
view	5.624e+04	1754.239	32.059	0.000	5.28e+04	5.97e+04

condition	2.728e+04	1941.072	14.055	0.000	2.35e+04	3.11e+04
grade	5.842e+04	1841.547	31.726	0.000	5.48e+04	6.2e+04
sqft_above	108.0880	14.578	7.414	0.000	79.514	136.662
sqft_basement	56.8494	15.084	3.769	0.000	27.284	86.415
yr_built	-733.8554	64.750	-11.334	0.000	-860.770	-606.940
yr_renovated	2913.6565	384.006	7.588	0.000	2160.976	3666.337
lat	1.904e+05	6.39e+04	2.981	0.003	6.52e+04	3.16e+05
long	-1.415e+05	4.58e+04	-3.087	0.002	-2.31e+05	-5.17e+04
sqft_living15	10.9413	2.911	3.758	0.000	5.235	16.648
sqft_lot15	-0.1623	0.061	-2.655	0.008	-0.282	-0.042
renovated	-5.776e+06	7.66e+05	-7.536	0.000	-7.28e+06	-4.27e+06
has_basement	-2.864e+04	4319.474	-6.630	0.000	-3.71e+04	-2.02e+04
zipcode_98002	3.22e+04	1.46e+04	2.213	0.027	3673.608	6.07e+04
zipcode_98003	-2.569e+04	1.3e+04	-1.974	0.048	-5.12e+04	-178.813
zipcode_98004	7.188e+05	2.36e+04	30.396	0.000	6.72e+05	7.65e+05
zipcode_98005	2.538e+05	2.53e+04	10.030	0.000	2.04e+05	3.03e+05
zipcode_98006	2.305e+05	2.07e+04	11.149	0.000	1.9e+05	2.71e+05
zipcode_98007	1.958e+05	2.61e+04	7.501	0.000	1.45e+05	2.47e+05
zipcode_98008	2.069e+05	2.48e+04	8.348	0.000	1.58e+05	2.56e+05
zipcode_98010	1.031e+05	2.22e+04	4.647	0.000	5.96e+04	1.47e+05
zipcode_98011	4.488e+04	3.23e+04	1.391	0.164	-1.83e+04	1.08e+05
zipcode_98014	9.37e+04	3.55e+04	2.641	0.008	2.42e+04	1.63e+05
zipcode_98019	4.642e+04	3.5e+04	1.327	0.184	-2.21e+04	1.15e+05
zipcode_98022	3.912e+04	1.93e+04	2.026	0.043	1276.269	7.7e+04
zipcode_98023	-4.584e+04	1.2e+04	-3.829	0.000	-6.93e+04	-2.24e+04
zipcode_98024	1.526e+05	3.12e+04	4.897	0.000	9.15e+04	2.14e+05
zipcode_98027	1.612e+05	2.12e+04	7.590	0.000	1.2e+05	2.03e+05
zipcode_98028	3.675e+04	3.13e+04	1.173	0.241	-2.46e+04	9.81e+04
zipcode_98029	1.972e+05	2.42e+04	8.132	0.000	1.5e+05	2.45e+05
zipcode_98030	636.3874	1.43e+04	0.045	0.965	-2.74e+04	2.87e+04
zipcode_98031	4600.7954	1.49e+04	0.309	0.758	-2.46e+04	3.38e+04
zipcode_98032	-6833.3457	1.73e+04	-0.395	0.693	-4.07e+04	2.71e+04
zipcode_98033	2.996e+05	2.69e+04	11.149	0.000	2.47e+05	3.52e+05
zipcode_98034	1.327e+05	2.88e+04	4.605	0.000	7.62e+04	1.89e+05
zipcode_98038	4.834e+04	1.61e+04	3.006	0.003	1.68e+04	7.99e+04

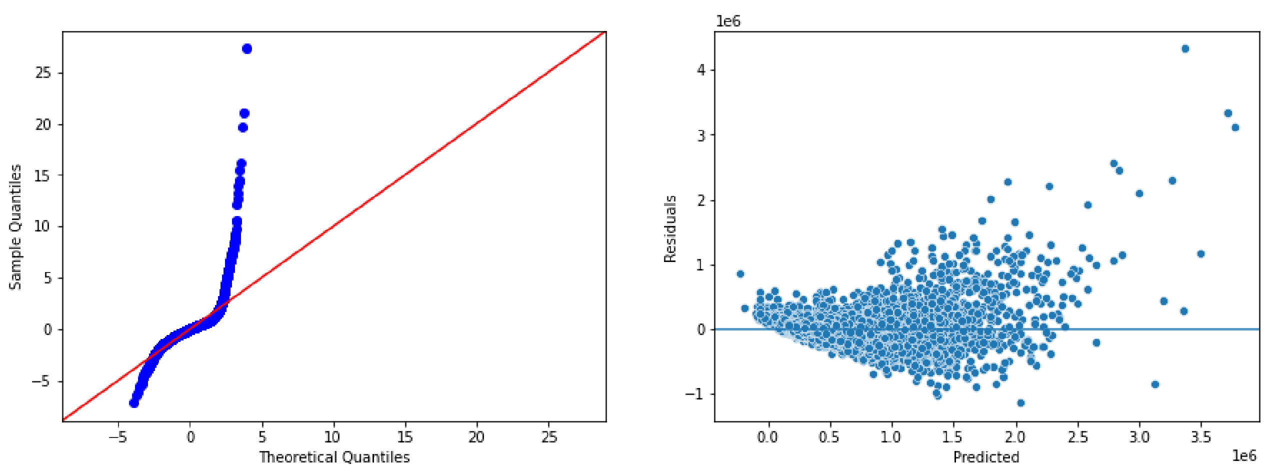
zipcode_98039	1.244e+06	3.2e+04	38.860	0.000	1.18e+06	1.31e+06
zipcode_98040	4.604e+05	2.09e+04	22.018	0.000	4.19e+05	5.01e+05
zipcode_98042	1.096e+04	1.37e+04	0.800	0.424	-1.59e+04	3.78e+04
zipcode_98045	1.317e+05	2.97e+04	4.433	0.000	7.34e+04	1.9e+05
zipcode_98052	1.751e+05	2.74e+04	6.380	0.000	1.21e+05	2.29e+05
zipcode_98053	1.536e+05	2.94e+04	5.227	0.000	9.6e+04	2.11e+05
zipcode_98055	2.392e+04	1.66e+04	1.441	0.150	-8618.375	5.65e+04
zipcode_98056	6.491e+04	1.8e+04	3.598	0.000	2.95e+04	1e+05
zipcode_98058	1.597e+04	1.57e+04	1.018	0.309	-1.48e+04	4.67e+04
zipcode_98059	6.302e+04	1.77e+04	3.563	0.000	2.84e+04	9.77e+04
zipcode_98065	9.525e+04	2.74e+04	3.478	0.001	4.16e+04	1.49e+05
zipcode_98070	-5.57e+04	2.09e+04	-2.671	0.008	-9.66e+04	-1.48e+04
zipcode_98072	8.25e+04	3.21e+04	2.571	0.010	1.96e+04	1.45e+05
zipcode_98074	1.377e+05	2.6e+04	5.302	0.000	8.68e+04	1.89e+05
zipcode_98075	1.417e+05	2.5e+04	5.673	0.000	9.27e+04	1.91e+05
zipcode_98077	5.934e+04	3.34e+04	1.778	0.075	-6086.166	1.25e+05
zipcode_98092	-2.674e+04	1.3e+04	-2.053	0.040	-5.23e+04	-1209.094
zipcode_98102	4.501e+05	2.77e+04	16.235	0.000	3.96e+05	5.04e+05
zipcode_98103	2.62e+05	2.6e+04	10.080	0.000	2.11e+05	3.13e+05
zipcode_98105	4.016e+05	2.67e+04	15.033	0.000	3.49e+05	4.54e+05
zipcode_98106	9.711e+04	1.93e+04	5.042	0.000	5.94e+04	1.35e+05
zipcode_98107	2.667e+05	2.68e+04	9.950	0.000	2.14e+05	3.19e+05
zipcode_98108	7.986e+04	2.13e+04	3.752	0.000	3.81e+04	1.22e+05
zipcode_98109	4.222e+05	2.77e+04	15.266	0.000	3.68e+05	4.76e+05
zipcode_98112	5.576e+05	2.45e+04	22.756	0.000	5.1e+05	6.06e+05
zipcode_98115	2.586e+05	2.64e+04	9.782	0.000	2.07e+05	3.1e+05
zipcode_98116	2.261e+05	2.15e+04	10.512	0.000	1.84e+05	2.68e+05
zipcode_98117	2.343e+05	2.68e+04	8.754	0.000	1.82e+05	2.87e+05
zipcode_98118	1.281e+05	1.88e+04	6.820	0.000	9.13e+04	1.65e+05
zipcode_98119	4.035e+05	2.61e+04	15.448	0.000	3.52e+05	4.55e+05
zipcode_98122	2.824e+05	2.33e+04	12.138	0.000	2.37e+05	3.28e+05
zipcode_98125	1.243e+05	2.85e+04	4.355	0.000	6.83e+04	1.8e+05
zipcode_98126	1.408e+05	1.98e+04	7.127	0.000	1.02e+05	1.8e+05
zipcode_98133	7.636e+04	2.95e+04	2.591	0.010	1.86e+04	1.34e+05
zipcode_98136	1.903e+05	2.02e+04	9.401	0.000	1.51e+05	2.3e+05

zipcode_98144	2.232e+05	2.17e+04	10.306	0.000	1.81e+05	2.66e+05
zipcode_98146	6.589e+04	1.81e+04	3.650	0.000	3.05e+04	1.01e+05
zipcode_98148	4.184e+04	2.45e+04	1.705	0.088	-6261.188	8.99e+04
zipcode_98155	5.779e+04	3.06e+04	1.887	0.059	-2239.720	1.18e+05
zipcode_98166	1.62e+04	1.65e+04	0.980	0.327	-1.62e+04	4.86e+04
zipcode_98168	4.138e+04	1.75e+04	2.367	0.018	7109.444	7.57e+04
zipcode_98177	1.196e+05	3.08e+04	3.889	0.000	5.93e+04	1.8e+05
zipcode_98178	7276.2285	1.81e+04	0.403	0.687	-2.81e+04	4.27e+04
zipcode_98188	8379.1848	1.85e+04	0.453	0.651	-2.79e+04	4.47e+04
zipcode_98198	-2.382e+04	1.4e+04	-1.697	0.090	-5.13e+04	3687.116
zipcode_98199	3.044e+05	2.54e+04	11.964	0.000	2.55e+05	3.54e+05

Omnibus:	20806.796	**Durbin-Watson:**	1.995
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4281101.044
Skew:	4.158	**Prob(JB):**	0.00
Kurtosis:	71.471	**Cond. No.**	3.07e+13

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.07e+13. This might indicate that there are
strong multicollinearity or other numerical problems.

```
Out[33]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1c35bdb5f10>
```



Our baseline model's residuals does not meet the normality or the homoscedasticity assumption yet. We need to consider multicollinearity and address the columns that may be compromising the model. Additionally, we need to address outliers.

Multicollinearity Check

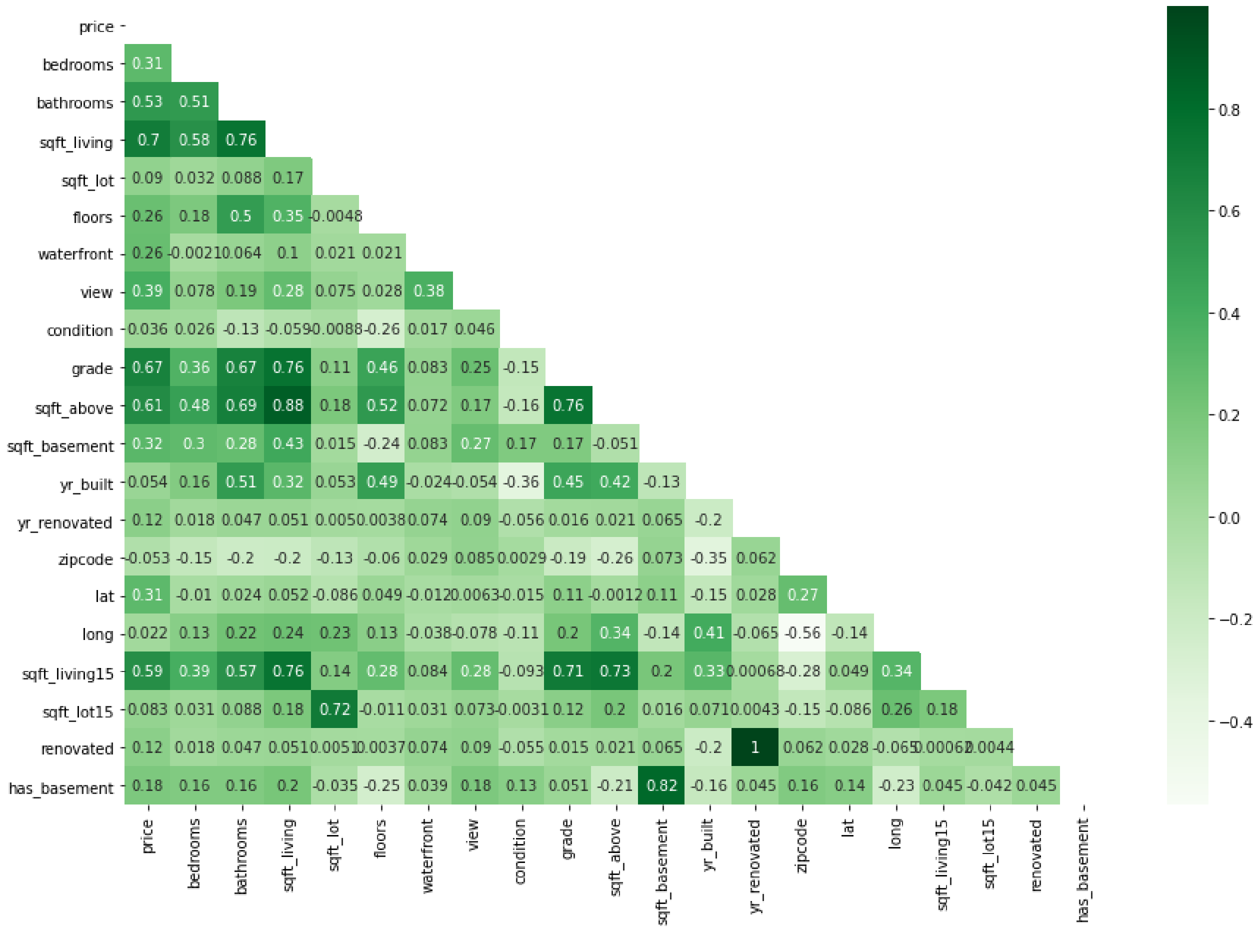
Correlation Matrices - Before

It looks like there is multicollinearity between the different columns. To address these I will be creating matrices and removing columns that may be causing the issues.

In [34]:
```
df.drop(['id', 'date'], axis=1, inplace=True)
```

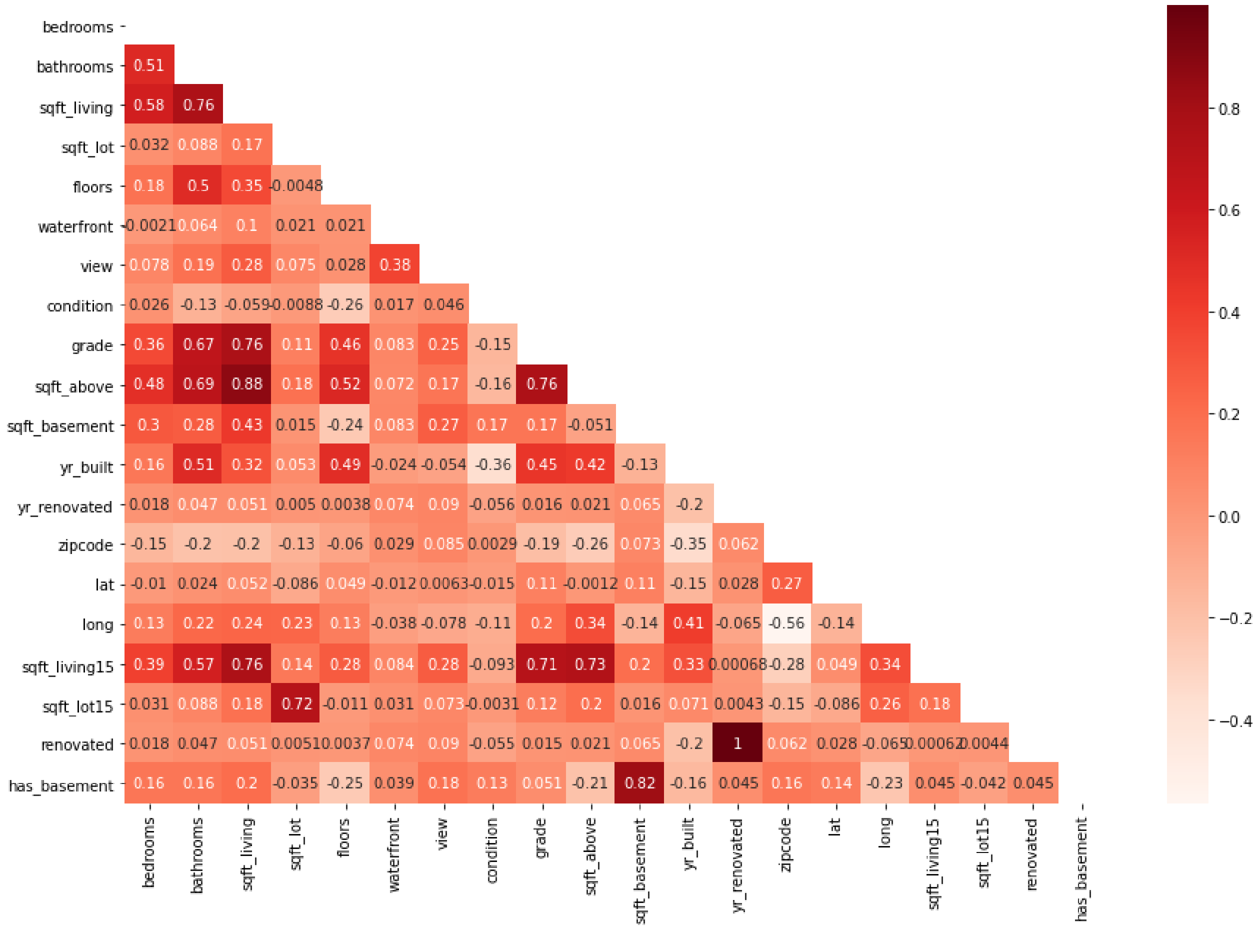
In [35]:
```
mask = np.zeros_like(df.corr())
mask[np.triu_indices_from(mask)] = True
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(df.corr(),annot=True, mask=mask, cmap='Greens')
```

Out[35]: <AxesSubplot:>



In [36]:
```
mask = np.zeros_like(df.drop('price', axis=1).corr())
mask[np.triu_indices_from(mask)] = True
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(df.drop('price', axis=1).corr(),annot=True, mask=mask, cmap='Reds')
```

Out[36]: <AxesSubplot:>



```
In [37]: df.corr()['price'].abs().sort_values(ascending=False)
```

```
Out[37]: price            1.000000
         sqft_living      0.701917
         grade            0.667951
         sqft_above       0.605368
         sqft_living15    0.585241
         bathrooms        0.525906
         view             0.393497
         sqft_basement    0.321108
         bedrooms         0.308787
         lat              0.306692
         waterfront       0.264306
         floors           0.256804
         has_basement     0.178264
         yr_renovated     0.117855
         renovated        0.117543
         sqft_lot         0.089876
         sqft_lot15       0.082845
         yr_built         0.053953
         zipcode          0.053402
         condition        0.036056
         long             0.022036
         Name: price, dtype: float64
```

Looking at the correlation matrix of the original df (prior to being one hot encoded for simplicity of the visual), there seems to be strong correlations between a multitude of parameters when we take our cut-off point as 0.75. Dropping the 'sqft_living' column will allow us to clear almost all the strong correlations. We are additionally not losing meaningful data since this column is equal to the sum of 'sqft_above' and 'sqft_basement'.

In [38]:
```
df.drop('sqft_living', axis=1, inplace=True)
```

Since we have engineered a feature called 'has_basement' using 'sqft_basement' there is a high correlation between these columns as well. Since we are still keeping the basement information to a certain degree, we can also drop 'sqft_basement'

In [39]:
```
df.drop('sqft_basement', axis=1, inplace=True)
```

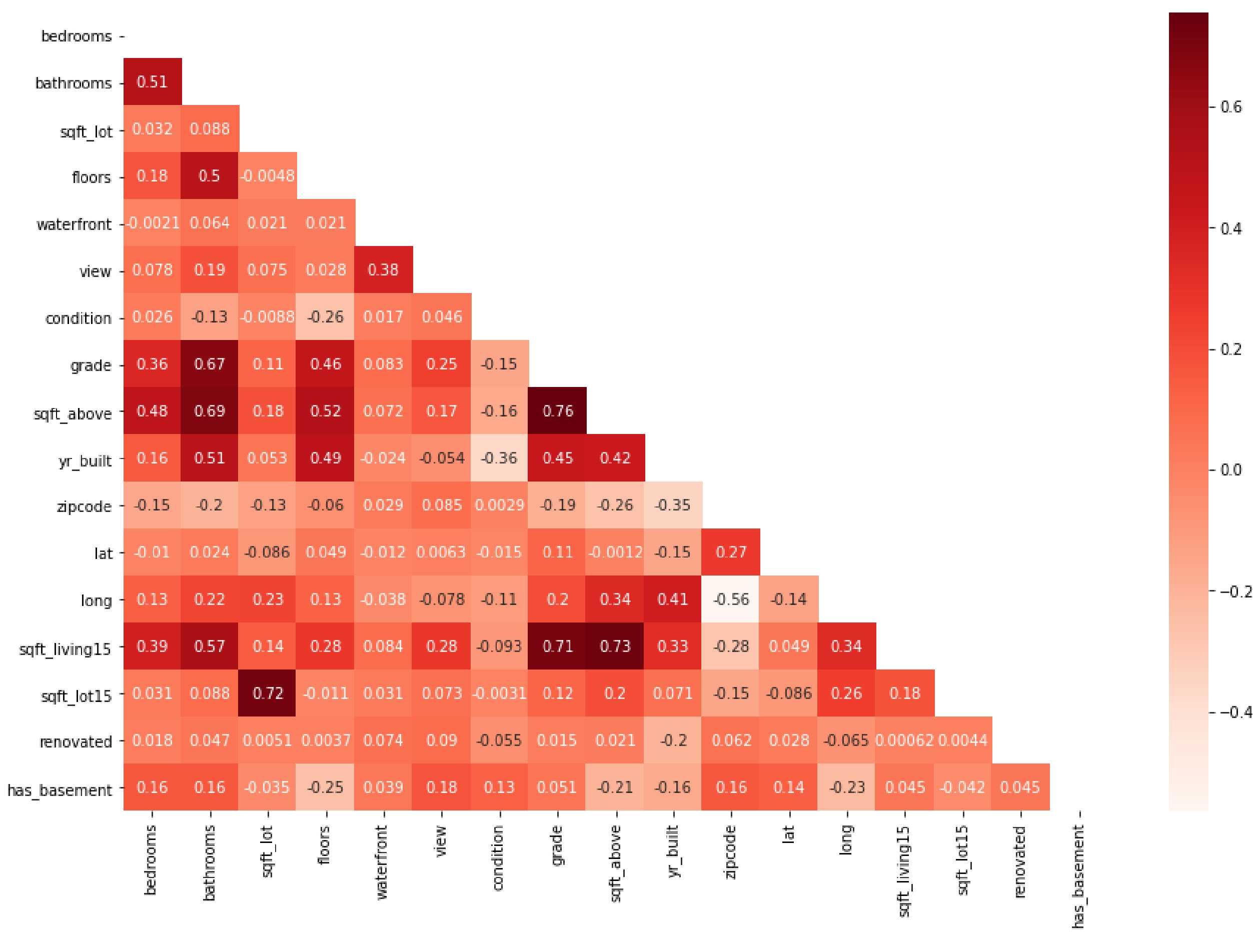
Similar to above, since we have engineered the column 'renovated' using 'yr_renovated', we can also drop the 'yr_renovated' column.

In [40]:
```
df.drop('yr_renovated', axis=1, inplace=True)
```

Correlation Matrix - After

In [41]:
```
mask = np.zeros_like(df.drop('price', axis=1).corr())
mask[np.triu_indices_from(mask)] = True
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(df.drop('price', axis=1).corr(),annot=True, mask=mask, cmap='Reds')
```

Out[41]:
```
<AxesSubplot:>
```



Our correlation matrix looks much better with the exception of 'sqft_above' and 'grade'. As this value is so close to our cut-off value of 0.75, we will be keeping it in to see if it causes any additional issues after adjustments to the model.

MODEL

For this project we were asked to specifically use multiple linear regression (MLR) so our model will be an MLR model.

In [42]:
```
drop_cols = ['id', 'date', 'sqft_living', 'sqft_basement', 'yr_renovated']
df_ohe.drop(drop_cols, axis=1, inplace=True)
```

In [43]:
```
model_lin_reg(df=df_ohe)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.800
Model:	OLS	**Adj. R-squared:**	0.799
Method:	Least Squares	**F-statistic:**	1010.
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:00	**Log-Likelihood:**	-2.9003e+05
No. Observations:	21597	**AIC:**	5.802e+05
Df Residuals:	21511	**BIC:**	5.809e+05
Df Model:	85		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.422e+07	6.3e+06	-3.845	0.000	-3.66e+07	-1.19e+07
bedrooms	-1.763e+04	1544.455	-11.413	0.000	-2.07e+04	-1.46e+04
bathrooms	4.377e+04	2643.295	16.557	0.000	3.86e+04	4.89e+04
sqft_lot	0.2560	0.039	6.499	0.000	0.179	0.333
floors	-6.367e+04	3189.871	-19.961	0.000	-6.99e+04	-5.74e+04
waterfront	6.937e+05	1.51e+04	46.012	0.000	6.64e+05	7.23e+05
view	6.292e+04	1770.266	35.543	0.000	5.95e+04	6.64e+04
condition	2.922e+04	1965.696	14.866	0.000	2.54e+04	3.31e+04
grade	6.338e+04	1864.517	33.994	0.000	5.97e+04	6.7e+04
sqft_above	190.3529	3.131	60.789	0.000	184.215	196.491
yr_built	-861.1302	65.764	-13.094	0.000	-990.033	-732.228
lat	2.032e+05	6.5e+04	3.125	0.002	7.58e+04	3.31e+05
long	-1.284e+05	4.67e+04	-2.749	0.006	-2.2e+05	-3.69e+04
sqft_living15	21.3812	2.944	7.264	0.000	15.612	27.151
sqft_lot15	-0.0880	0.062	-1.418	0.156	-0.210	0.034
renovated	4.037e+04	6547.392	6.166	0.000	2.75e+04	5.32e+04
has_basement	6.225e+04	3093.061	20.124	0.000	5.62e+04	6.83e+04

zipcode_98002	3.318e+04	1.48e+04	2.236	0.025	4093.329	6.23e+04
zipcode_98003	-2.615e+04	1.33e+04	-1.971	0.049	-5.22e+04	-149.702
zipcode_98004	7.234e+05	2.41e+04	30.009	0.000	6.76e+05	7.71e+05
zipcode_98005	2.487e+05	2.58e+04	9.653	0.000	1.98e+05	2.99e+05
zipcode_98006	2.355e+05	2.11e+04	11.178	0.000	1.94e+05	2.77e+05
zipcode_98007	1.937e+05	2.66e+04	7.286	0.000	1.42e+05	2.46e+05
zipcode_98008	2.025e+05	2.53e+04	8.018	0.000	1.53e+05	2.52e+05
zipcode_98010	9.554e+04	2.26e+04	4.223	0.000	5.12e+04	1.4e+05
zipcode_98011	3.791e+04	3.29e+04	1.154	0.249	-2.65e+04	1.02e+05
zipcode_98014	8.408e+04	3.61e+04	2.330	0.020	1.33e+04	1.55e+05
zipcode_98019	4.459e+04	3.56e+04	1.253	0.210	-2.52e+04	1.14e+05
zipcode_98022	3.069e+04	1.97e+04	1.562	0.118	-7832.522	6.92e+04
zipcode_98023	-4.989e+04	1.22e+04	-4.088	0.000	-7.38e+04	-2.6e+04
zipcode_98024	1.431e+05	3.18e+04	4.504	0.000	8.08e+04	2.05e+05
zipcode_98027	1.547e+05	2.16e+04	7.153	0.000	1.12e+05	1.97e+05
zipcode_98028	3.521e+04	3.19e+04	1.103	0.270	-2.73e+04	9.77e+04
zipcode_98029	1.881e+05	2.47e+04	7.614	0.000	1.4e+05	2.36e+05
zipcode_98030	-1278.3007	1.46e+04	-0.088	0.930	-2.99e+04	2.73e+04
zipcode_98031	2733.1401	1.52e+04	0.180	0.857	-2.7e+04	3.25e+04
zipcode_98032	-6121.9357	1.76e+04	-0.347	0.728	-4.07e+04	2.84e+04
zipcode_98033	2.972e+05	2.74e+04	10.857	0.000	2.44e+05	3.51e+05
zipcode_98034	1.22e+05	2.93e+04	4.157	0.000	6.45e+04	1.8e+05
zipcode_98038	4.76e+04	1.64e+04	2.906	0.004	1.55e+04	7.97e+04
zipcode_98039	1.254e+06	3.26e+04	38.492	0.000	1.19e+06	1.32e+06
zipcode_98040	4.697e+05	2.13e+04	22.035	0.000	4.28e+05	5.12e+05
zipcode_98042	9795.9407	1.4e+04	0.702	0.483	-1.76e+04	3.72e+04
zipcode_98045	1.186e+05	3.03e+04	3.920	0.000	5.93e+04	1.78e+05
zipcode_98052	1.633e+05	2.79e+04	5.844	0.000	1.09e+05	2.18e+05
zipcode_98053	1.436e+05	2.99e+04	4.794	0.000	8.49e+04	2.02e+05
zipcode_98055	2.336e+04	1.69e+04	1.381	0.167	-9798.967	5.65e+04
zipcode_98056	6.749e+04	1.84e+04	3.672	0.000	3.15e+04	1.04e+05
zipcode_98058	1.286e+04	1.6e+04	0.805	0.421	-1.85e+04	4.42e+04
zipcode_98059	5.588e+04	1.8e+04	3.099	0.002	2.05e+04	9.12e+04
zipcode_98065	8.765e+04	2.79e+04	3.142	0.002	3.3e+04	1.42e+05
zipcode_98070	-5.806e+04	2.13e+04	-2.730	0.006	-9.97e+04	-1.64e+04

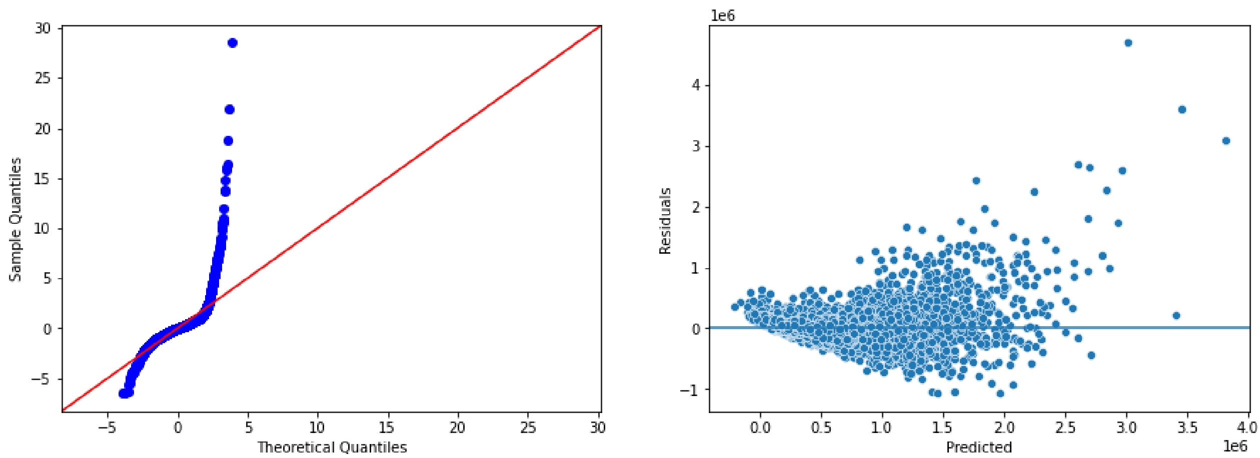
zipcode_98072	7.39e+04	3.27e+04	2.261	0.024	9836.718	1.38e+05
zipcode_98074	1.277e+05	2.65e+04	4.825	0.000	7.58e+04	1.8e+05
zipcode_98075	1.295e+05	2.54e+04	5.089	0.000	7.96e+04	1.79e+05
zipcode_98077	4.529e+04	3.4e+04	1.332	0.183	-2.14e+04	1.12e+05
zipcode_98092	-2.867e+04	1.33e+04	-2.161	0.031	-5.47e+04	-2667.678
zipcode_98102	4.422e+05	2.83e+04	15.646	0.000	3.87e+05	4.98e+05
zipcode_98103	2.628e+05	2.65e+04	9.927	0.000	2.11e+05	3.15e+05
zipcode_98105	3.882e+05	2.72e+04	14.279	0.000	3.35e+05	4.41e+05
zipcode_98106	9.546e+04	1.96e+04	4.866	0.000	5.7e+04	1.34e+05
zipcode_98107	2.629e+05	2.73e+04	9.634	0.000	2.09e+05	3.16e+05
zipcode_98108	8.08e+04	2.17e+04	3.731	0.000	3.84e+04	1.23e+05
zipcode_98109	4.174e+05	2.81e+04	14.844	0.000	3.62e+05	4.73e+05
zipcode_98112	5.539e+05	2.5e+04	22.191	0.000	5.05e+05	6.03e+05
zipcode_98115	2.56e+05	2.69e+04	9.511	0.000	2.03e+05	3.09e+05
zipcode_98116	2.243e+05	2.19e+04	10.240	0.000	1.81e+05	2.67e+05
zipcode_98117	2.321e+05	2.73e+04	8.515	0.000	1.79e+05	2.85e+05
zipcode_98118	1.313e+05	1.91e+04	6.862	0.000	9.38e+04	1.69e+05
zipcode_98119	3.943e+05	2.66e+04	14.841	0.000	3.42e+05	4.46e+05
zipcode_98122	2.704e+05	2.37e+04	11.410	0.000	2.24e+05	3.17e+05
zipcode_98125	1.21e+05	2.91e+04	4.164	0.000	6.4e+04	1.78e+05
zipcode_98126	1.369e+05	2.01e+04	6.804	0.000	9.74e+04	1.76e+05
zipcode_98133	7.66e+04	3e+04	2.553	0.011	1.78e+04	1.35e+05
zipcode_98136	1.855e+05	2.06e+04	8.994	0.000	1.45e+05	2.26e+05
zipcode_98144	2.254e+05	2.2e+04	10.231	0.000	1.82e+05	2.69e+05
zipcode_98146	6.678e+04	1.84e+04	3.628	0.000	3.07e+04	1.03e+05
zipcode_98148	3.46e+04	2.5e+04	1.382	0.167	-1.45e+04	8.37e+04
zipcode_98155	5.55e+04	3.12e+04	1.779	0.075	-5656.754	1.17e+05
zipcode_98166	1.449e+04	1.68e+04	0.860	0.390	-1.85e+04	4.75e+04
zipcode_98168	4.58e+04	1.78e+04	2.573	0.010	1.09e+04	8.07e+04
zipcode_98177	1.168e+05	3.13e+04	3.727	0.000	5.54e+04	1.78e+05
zipcode_98178	1.105e+04	1.84e+04	0.601	0.548	-2.5e+04	4.71e+04
zipcode_98188	1.379e+04	1.89e+04	0.731	0.465	-2.32e+04	5.08e+04
zipcode_98198	-2.15e+04	1.43e+04	-1.504	0.133	-4.95e+04	6528.999
zipcode_98199	3.025e+05	2.59e+04	11.685	0.000	2.52e+05	3.53e+05

Omnibus:	21922.754	**Durbin-Watson:**	1.992

Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5315892.814
Skew:	4.513	**Prob(JB):**	0.00
Kurtosis:	79.328	**Cond. No.**	2.84e+08

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.84e+08. This might indicate that there are
strong multicollinearity or other numerical problems.

Out[43]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1c35b158670>



The sqft_lot15 p-value is higher than our alpha of 0.05 meaning that this coefficient is not statistically significant, so we can go ahead and drop this parameter.

In [44]:
```
df_ohe.drop('sqft_lot15', axis=1, inplace=True)
```

We also wanted to create a feature to see if having a larger sqft (excluding the basement) would have any effect on the sale price of the home.

In [45]:
```
df_ohe['has_larger_sqft_than_neighbors'] = df_ohe['sqft_living15'] < df_ohe['sqft_above
df_ohe['has_larger_sqft_than_neighbors'] = df_ohe['has_larger_sqft_than_neighbors'].ast
df_ohe.drop('sqft_living15', axis=1, inplace=True)
df_ohe.head()
```

Out[45]:

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	yı
0	221900.0	3	1.00	5650	1.0	0.0	0.0	3	7	1180	
1	538000.0	3	2.25	7242	2.0	0.0	0.0	3	7	2170	
2	180000.0	2	1.00	10000	1.0	0.0	0.0	3	6	770	
3	604000.0	4	3.00	5000	1.0	0.0	0.0	5	7	1050	
4	510000.0	3	2.00	8080	1.0	0.0	0.0	3	8	1680	

5 rows × 85 columns

In [46]:
```
model_lin_reg(df=df_ohe)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.800
Model:	OLS	**Adj. R-squared:**	0.800
Method:	Least Squares	**F-statistic:**	1027.
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:01	**Log-Likelihood:**	-2.8999e+05
No. Observations:	21597	**AIC:**	5.801e+05
Df Residuals:	21512	**BIC:**	5.808e+05
Df Model:	84		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.425e+07	6.28e+06	-3.863	0.000	-3.66e+07	-1.19e+07
bedrooms	-1.68e+04	1541.928	-10.899	0.000	-1.98e+04	-1.38e+04
bathrooms	4.332e+04	2638.007	16.422	0.000	3.82e+04	4.85e+04
sqft_lot	0.2141	0.030	7.166	0.000	0.156	0.273
floors	-6.274e+04	3179.284	-19.734	0.000	-6.9e+04	-5.65e+04
waterfront	6.911e+05	1.5e+04	45.966	0.000	6.62e+05	7.21e+05
view	6.39e+04	1737.190	36.784	0.000	6.05e+04	6.73e+04
condition	2.946e+04	1960.975	15.022	0.000	2.56e+04	3.33e+04
grade	6.282e+04	1829.256	34.343	0.000	5.92e+04	6.64e+04
sqft_above	213.2508	3.187	66.919	0.000	207.005	219.497
yr_built	-881.6538	65.484	-13.464	0.000	-1010.006	-753.301
lat	1.978e+05	6.49e+04	3.048	0.002	7.06e+04	3.25e+05
long	-1.312e+05	4.65e+04	-2.819	0.005	-2.22e+05	-4e+04
renovated	4.136e+04	6531.960	6.332	0.000	2.86e+04	5.42e+04
has_basement	6.363e+04	3051.663	20.850	0.000	5.76e+04	6.96e+04
zipcode_98002	3.298e+04	1.48e+04	2.228	0.026	3963.897	6.2e+04
zipcode_98003	-2.736e+04	1.32e+04	-2.067	0.039	-5.33e+04	-1414.683
zipcode_98004	7.254e+05	2.4e+04	30.163	0.000	6.78e+05	7.72e+05
zipcode_98005	2.515e+05	2.57e+04	9.786	0.000	2.01e+05	3.02e+05
zipcode_98006	2.386e+05	2.1e+04	11.363	0.000	1.97e+05	2.8e+05
zipcode_98007	1.964e+05	2.65e+04	7.405	0.000	1.44e+05	2.48e+05
zipcode_98008	2.039e+05	2.52e+04	8.092	0.000	1.55e+05	2.53e+05
zipcode_98010	9.662e+04	2.26e+04	4.281	0.000	5.24e+04	1.41e+05

zipcode_98011	4.068e+04	3.28e+04	1.241	0.215	-2.36e+04	1.05e+05
zipcode_98014	8.781e+04	3.6e+04	2.438	0.015	1.72e+04	1.58e+05
zipcode_98019	4.763e+04	3.55e+04	1.341	0.180	-2.2e+04	1.17e+05
zipcode_98022	3.037e+04	1.96e+04	1.548	0.122	-8073.985	6.88e+04
zipcode_98023	-5.138e+04	1.22e+04	-4.219	0.000	-7.53e+04	-2.75e+04
zipcode_98024	1.409e+05	3.17e+04	4.448	0.000	7.88e+04	2.03e+05
zipcode_98027	1.551e+05	2.16e+04	7.190	0.000	1.13e+05	1.97e+05
zipcode_98028	3.818e+04	3.18e+04	1.199	0.231	-2.42e+04	1.01e+05
zipcode_98029	1.889e+05	2.46e+04	7.667	0.000	1.41e+05	2.37e+05
zipcode_98030	366.4616	1.46e+04	0.025	0.980	-2.82e+04	2.89e+04
zipcode_98031	1079.4560	1.52e+04	0.071	0.943	-2.86e+04	3.08e+04
zipcode_98032	-7082.3756	1.76e+04	-0.403	0.687	-4.16e+04	2.74e+04
zipcode_98033	2.993e+05	2.73e+04	10.957	0.000	2.46e+05	3.53e+05
zipcode_98034	1.23e+05	2.93e+04	4.199	0.000	6.56e+04	1.8e+05
zipcode_98038	4.753e+04	1.63e+04	2.909	0.004	1.55e+04	7.95e+04
zipcode_98039	1.254e+06	3.25e+04	38.575	0.000	1.19e+06	1.32e+06
zipcode_98040	4.749e+05	2.13e+04	22.348	0.000	4.33e+05	5.17e+05
zipcode_98042	9389.1280	1.39e+04	0.674	0.500	-1.79e+04	3.67e+04
zipcode_98045	1.176e+05	3.02e+04	3.896	0.000	5.84e+04	1.77e+05
zipcode_98052	1.65e+05	2.79e+04	5.919	0.000	1.1e+05	2.2e+05
zipcode_98053	1.455e+05	2.99e+04	4.869	0.000	8.69e+04	2.04e+05
zipcode_98055	2.435e+04	1.69e+04	1.443	0.149	-8737.813	5.74e+04
zipcode_98056	6.823e+04	1.83e+04	3.720	0.000	3.23e+04	1.04e+05
zipcode_98058	1.332e+04	1.59e+04	0.835	0.404	-1.79e+04	4.46e+04
zipcode_98059	5.721e+04	1.8e+04	3.182	0.001	2.2e+04	9.25e+04
zipcode_98065	8.945e+04	2.78e+04	3.217	0.001	3.5e+04	1.44e+05
zipcode_98070	-5.994e+04	2.11e+04	-2.847	0.004	-1.01e+05	-1.87e+04
zipcode_98072	7.646e+04	3.26e+04	2.345	0.019	1.25e+04	1.4e+05
zipcode_98074	1.308e+05	2.64e+04	4.956	0.000	7.91e+04	1.83e+05
zipcode_98075	1.329e+05	2.54e+04	5.243	0.000	8.32e+04	1.83e+05
zipcode_98077	4.759e+04	3.39e+04	1.403	0.161	-1.89e+04	1.14e+05
zipcode_98092	-2.821e+04	1.32e+04	-2.133	0.033	-5.41e+04	-2287.050
zipcode_98102	4.419e+05	2.82e+04	15.669	0.000	3.87e+05	4.97e+05
zipcode_98103	2.63e+05	2.64e+04	9.959	0.000	2.11e+05	3.15e+05
zipcode_98105	3.893e+05	2.71e+04	14.349	0.000	3.36e+05	4.42e+05

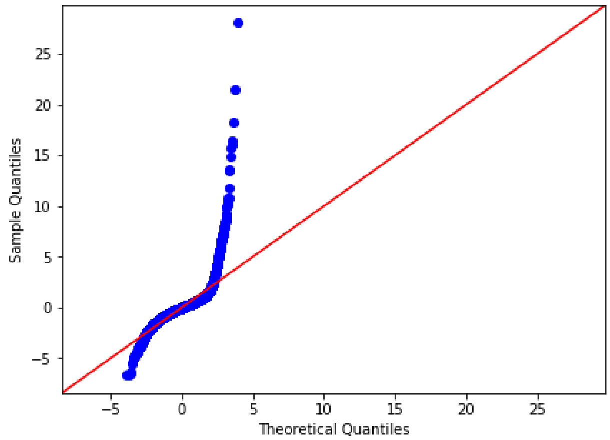
zipcode_98106	9.449e+04	1.96e+04	4.831	0.000	5.62e+04	1.33e+05
zipcode_98107	2.624e+05	2.72e+04	9.638	0.000	2.09e+05	3.16e+05
zipcode_98108	8.161e+04	2.16e+04	3.777	0.000	3.93e+04	1.24e+05
zipcode_98109	4.186e+05	2.81e+04	14.918	0.000	3.64e+05	4.74e+05
zipcode_98112	5.554e+05	2.49e+04	22.297	0.000	5.07e+05	6.04e+05
zipcode_98115	2.565e+05	2.69e+04	9.552	0.000	2.04e+05	3.09e+05
zipcode_98116	2.235e+05	2.18e+04	10.229	0.000	1.81e+05	2.66e+05
zipcode_98117	2.317e+05	2.72e+04	8.525	0.000	1.78e+05	2.85e+05
zipcode_98118	1.315e+05	1.91e+04	6.890	0.000	9.41e+04	1.69e+05
zipcode_98119	3.945e+05	2.65e+04	14.881	0.000	3.42e+05	4.46e+05
zipcode_98122	2.713e+05	2.36e+04	11.473	0.000	2.25e+05	3.18e+05
zipcode_98125	1.229e+05	2.9e+04	4.237	0.000	6.6e+04	1.8e+05
zipcode_98126	1.36e+05	2.01e+04	6.782	0.000	9.67e+04	1.75e+05
zipcode_98133	7.711e+04	2.99e+04	2.576	0.010	1.84e+04	1.36e+05
zipcode_98136	1.841e+05	2.06e+04	8.949	0.000	1.44e+05	2.24e+05
zipcode_98144	2.246e+05	2.2e+04	10.218	0.000	1.82e+05	2.68e+05
zipcode_98146	6.647e+04	1.84e+04	3.621	0.000	3.05e+04	1.02e+05
zipcode_98148	3.604e+04	2.5e+04	1.443	0.149	-1.29e+04	8.5e+04
zipcode_98155	5.694e+04	3.11e+04	1.829	0.067	-4090.436	1.18e+05
zipcode_98166	1.463e+04	1.68e+04	0.870	0.384	-1.83e+04	4.76e+04
zipcode_98168	4.695e+04	1.78e+04	2.644	0.008	1.21e+04	8.18e+04
zipcode_98177	1.174e+05	3.13e+04	3.757	0.000	5.62e+04	1.79e+05
zipcode_98178	1.199e+04	1.83e+04	0.654	0.513	-2.4e+04	4.8e+04
zipcode_98188	1.322e+04	1.88e+04	0.702	0.482	-2.37e+04	5.01e+04
zipcode_98198	-2.128e+04	1.43e+04	-1.491	0.136	-4.92e+04	6692.339
zipcode_98199	3.036e+05	2.58e+04	11.754	0.000	2.53e+05	3.54e+05
has_larger_sqft_than_neighbors	-3.716e+04	3131.183	-11.869	0.000	-4.33e+04	-3.1e+04

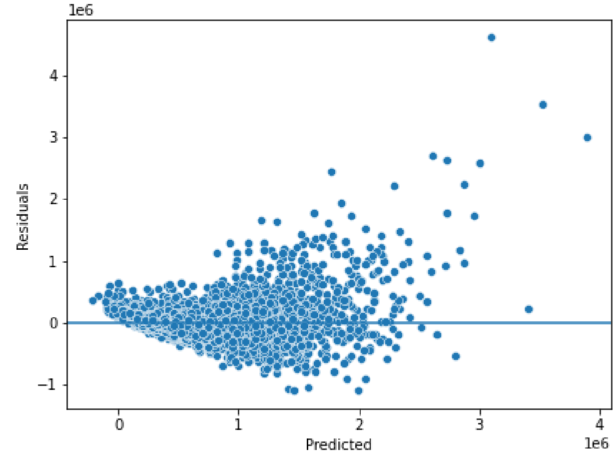
Omnibus:	21495.431	**Durbin-Watson:**	1.990
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4815678.728
Skew:	4.381	**Prob(JB):**	0.00
Kurtosis:	75.627	**Cond. No.**	2.47e+08

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.47e+08. This might indicate that there are strong multicollinearity or other numerical problems.

Out[46]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1c35da70370>





As we can see our residuals still seem to not be fitting the normality and homoscedasticity assumptions. To address this we can move on to the outlier removal process.

Outlier Removal

In [47]:
```
#Outlier Removal with the IQR method
#function snippet from Flatiron School Phase #2 Py Files.

def find_outliers_IQR(data):
    """Use Tukey's Method of outlier removal AKA InterQuartile-Range Rule
    and return boolean series where True indicates it is an outlier.
    - Calculates the range between the 75% and 25% quartiles
    - Outliers fall outside upper and lower limits, using a treshold of  1.5*IQR the 75

    IQR Range Calculation:
        res = df.describe()
        IQR = res['75%'] -  res['25%']
        lower_limit = res['25%'] - 1.5*IQR
        upper_limit = res['75%'] + 1.5*IQR

    Args:
        data (Series,or ndarray): data to test for outliers.

    Returns:
        [boolean Series]: A True/False for each row use to slice outliers.

    EXAMPLE USE:
    >> idx_outs = find_outliers_df(df['AdjustedCompensation'])
    >> good_data = df[~idx_outs].copy()

    """
    df_b=data
    res= df_b.describe()

    IQR = res['75%'] -  res['25%']
    lower_limit = res['25%'] - 1.5*IQR
    upper_limit = res['75%'] + 1.5*IQR
```

```
    idx_outs = (df_b>upper_limit) | (df_b<lower_limit)

    return idx_outs
```

In [48]:
```
#Making a copy of df_ohe for the second outlier removal process. Refer to next section.
df_IQR_price = df_ohe.copy()
df_IQR_price
```

Out[48]:

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_abov
0	221900.0	3	1.00	5650	1.0	0.0	0.0	3	7	118
1	538000.0	3	2.25	7242	2.0	0.0	0.0	3	7	217
2	180000.0	2	1.00	10000	1.0	0.0	0.0	3	6	77
3	604000.0	4	3.00	5000	1.0	0.0	0.0	5	7	105
4	510000.0	3	2.00	8080	1.0	0.0	0.0	3	8	168
...	...	...	...	...	...	...	...	...	...	
21592	360000.0	3	2.50	1131	3.0	0.0	0.0	3	8	153
21593	400000.0	4	2.50	5813	2.0	0.0	0.0	3	8	231
21594	402101.0	2	0.75	1350	2.0	0.0	0.0	3	7	102
21595	400000.0	3	2.50	2388	2.0	0.0	0.0	3	8	160
21596	325000.0	2	0.75	1076	2.0	0.0	0.0	3	7	102

21597 rows × 85 columns

Cleaning Outliers From All Numeric Columns

In [49]:
```
cols_to_check = ['price', 'bedrooms','bathrooms', 'sqft_lot', 'grade', 'sqft_above', 'c
```

In [50]:
```
for col in cols_to_check:
    df_ohe = df_ohe[find_outliers_IQR(df_ohe[col])==False]

df_ohe
```

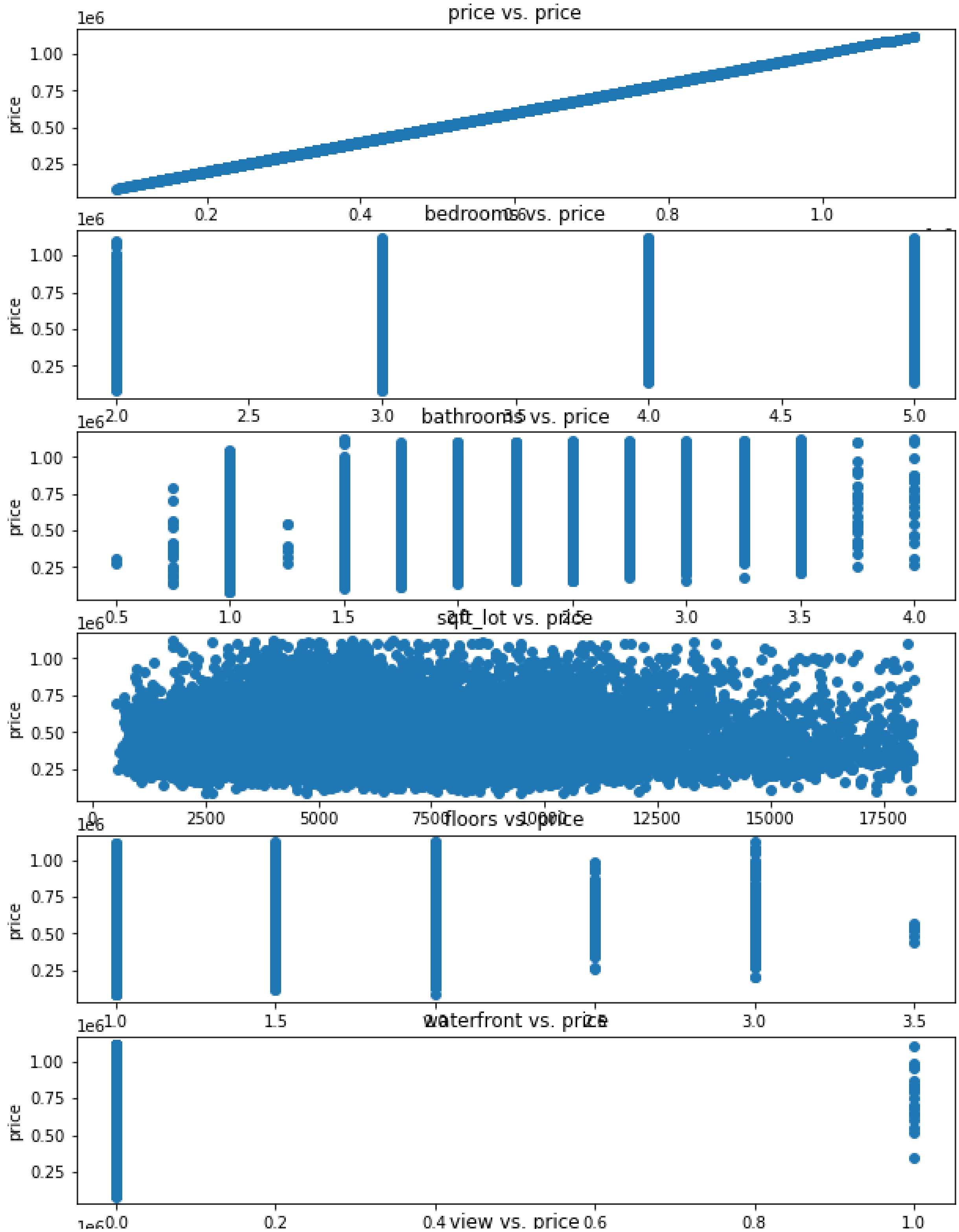
Out[50]:

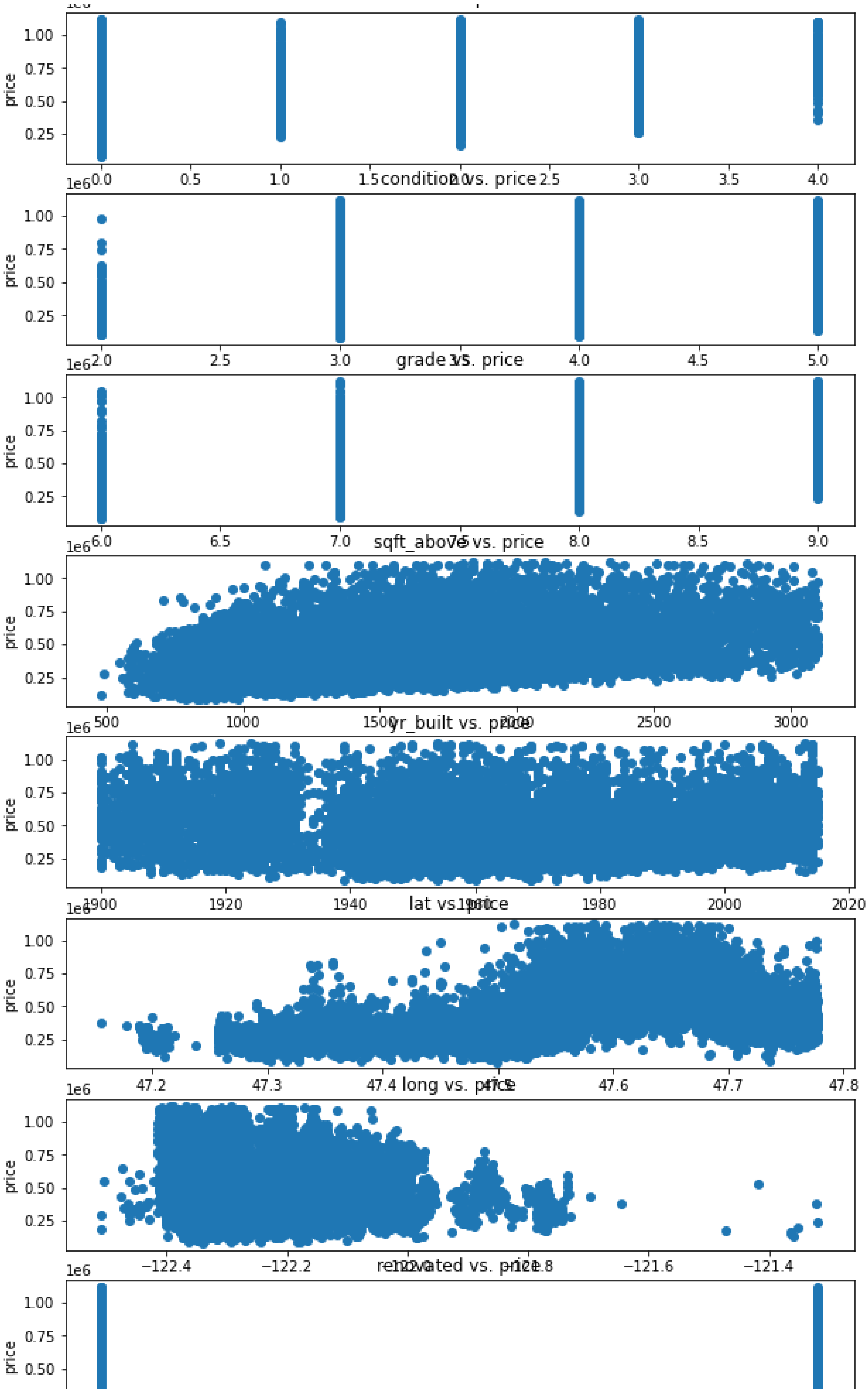
	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_abov
0	221900.0	3	1.00	5650	1.0	0.0	0.0	3	7	118
1	538000.0	3	2.25	7242	2.0	0.0	0.0	3	7	217
2	180000.0	2	1.00	10000	1.0	0.0	0.0	3	6	77
3	604000.0	4	3.00	5000	1.0	0.0	0.0	5	7	105
4	510000.0	3	2.00	8080	1.0	0.0	0.0	3	8	168
...	...	...	...	...	...	...	...	...	...	
21592	360000.0	3	2.50	1131	3.0	0.0	0.0	3	8	153
21593	400000.0	4	2.50	5813	2.0	0.0	0.0	3	8	231
21594	402101.0	2	0.75	1350	2.0	0.0	0.0	3	7	102

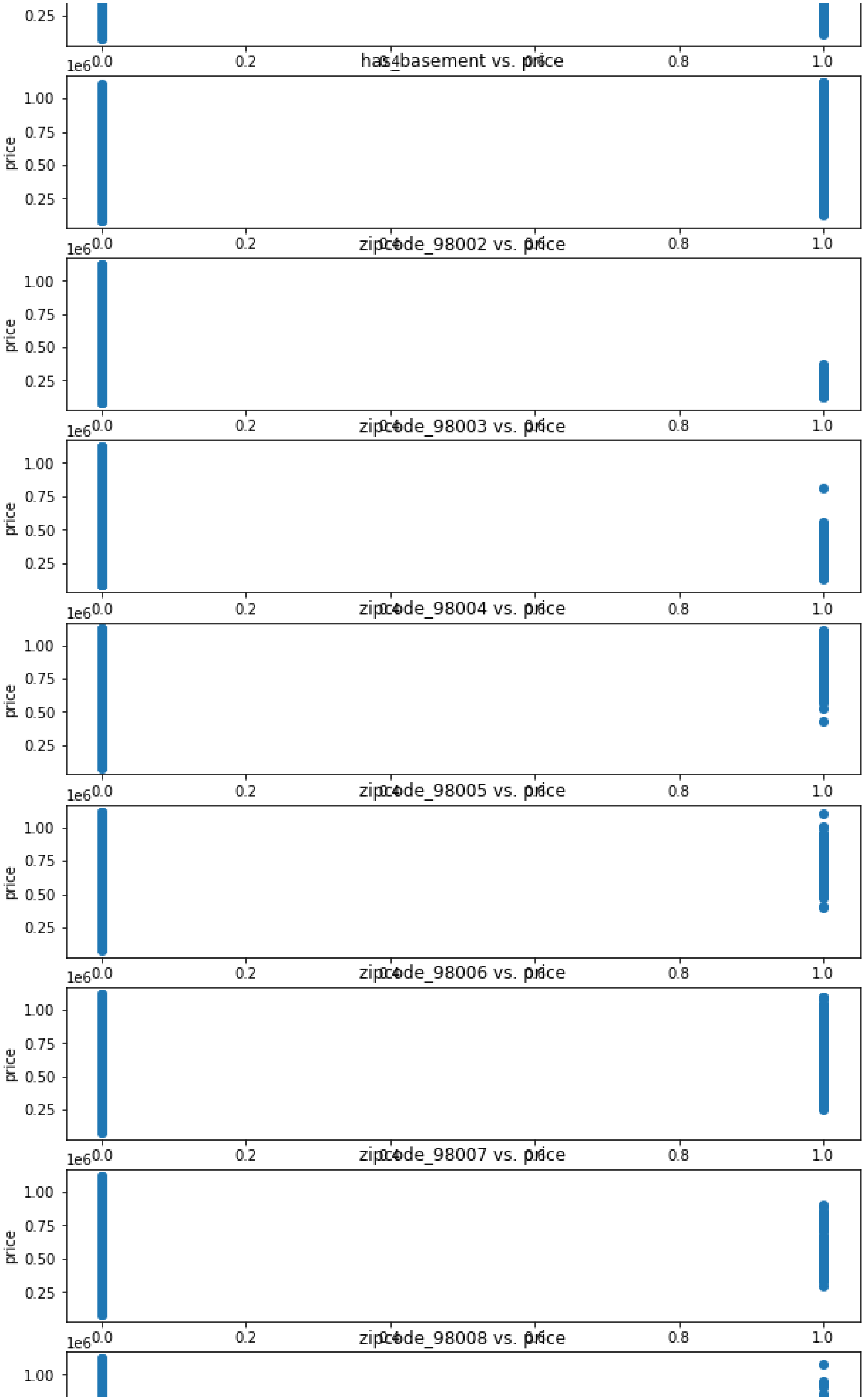
	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_abov
21595	400000.0	3	2.50	2388	2.0	0.0	0.0	3	8	160
21596	325000.0	2	0.75	1076	2.0	0.0	0.0	3	7	102

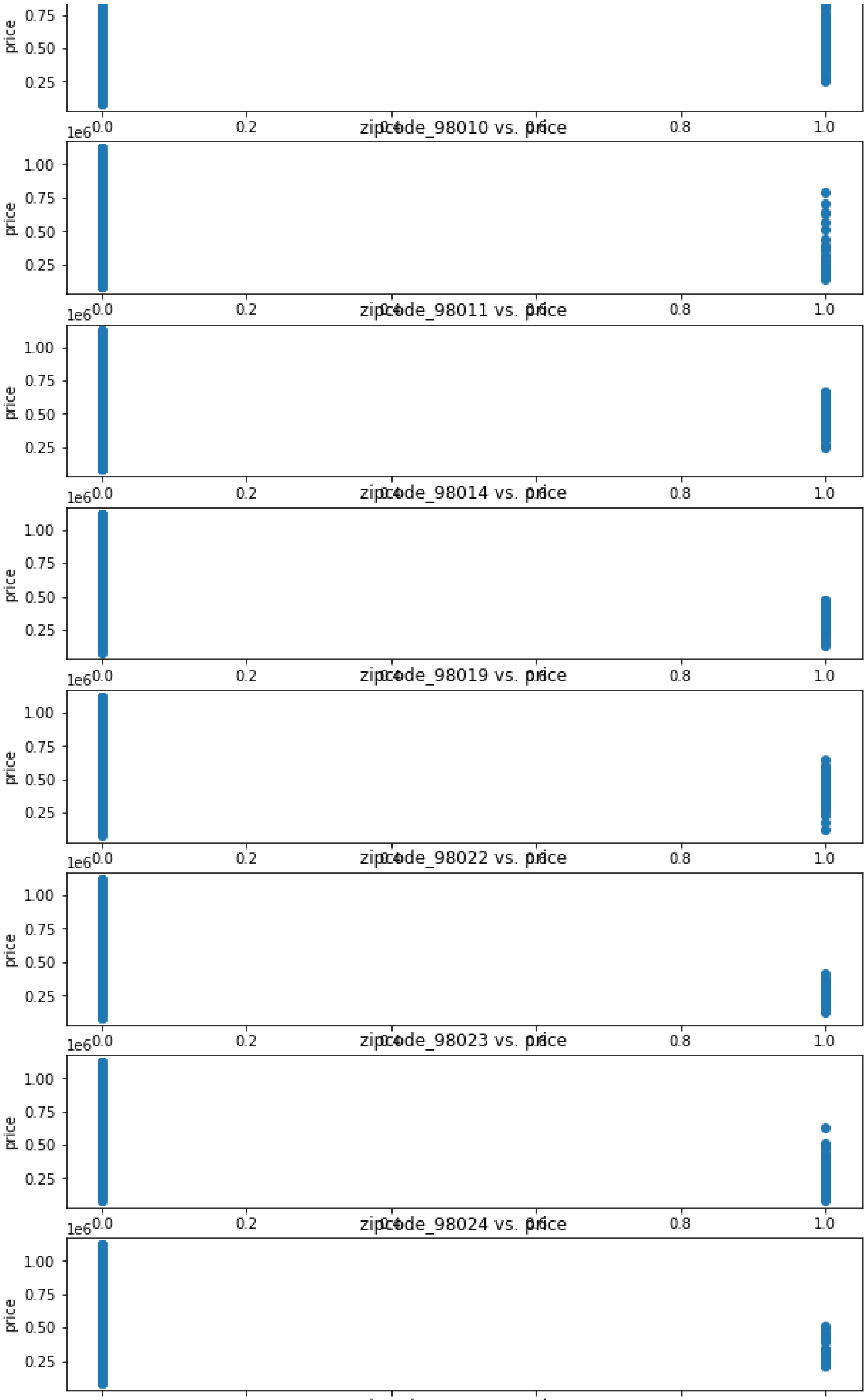
16556 rows × 85 columns

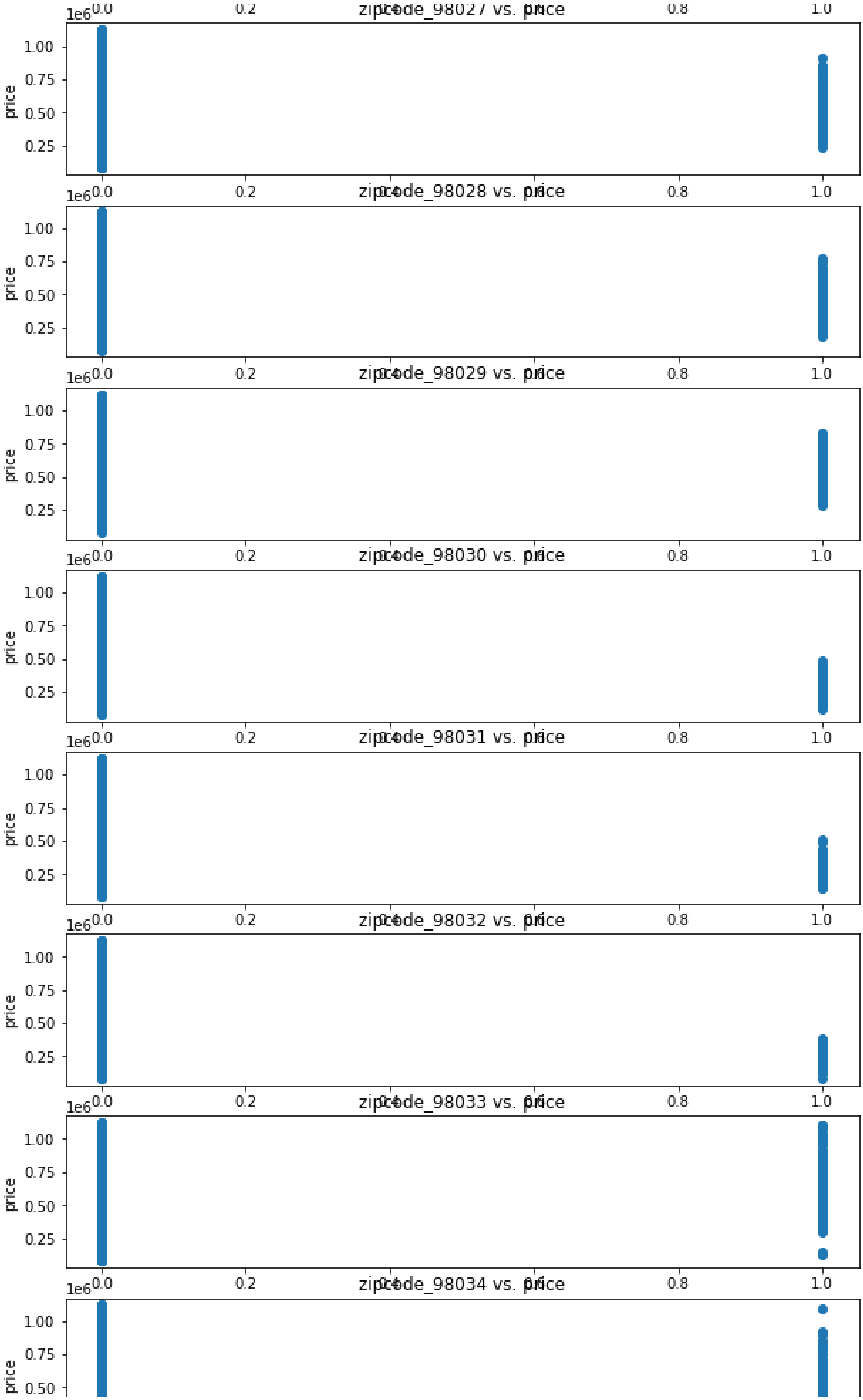
In [51]:
```
plot(df=df_ohe, target='price')
```

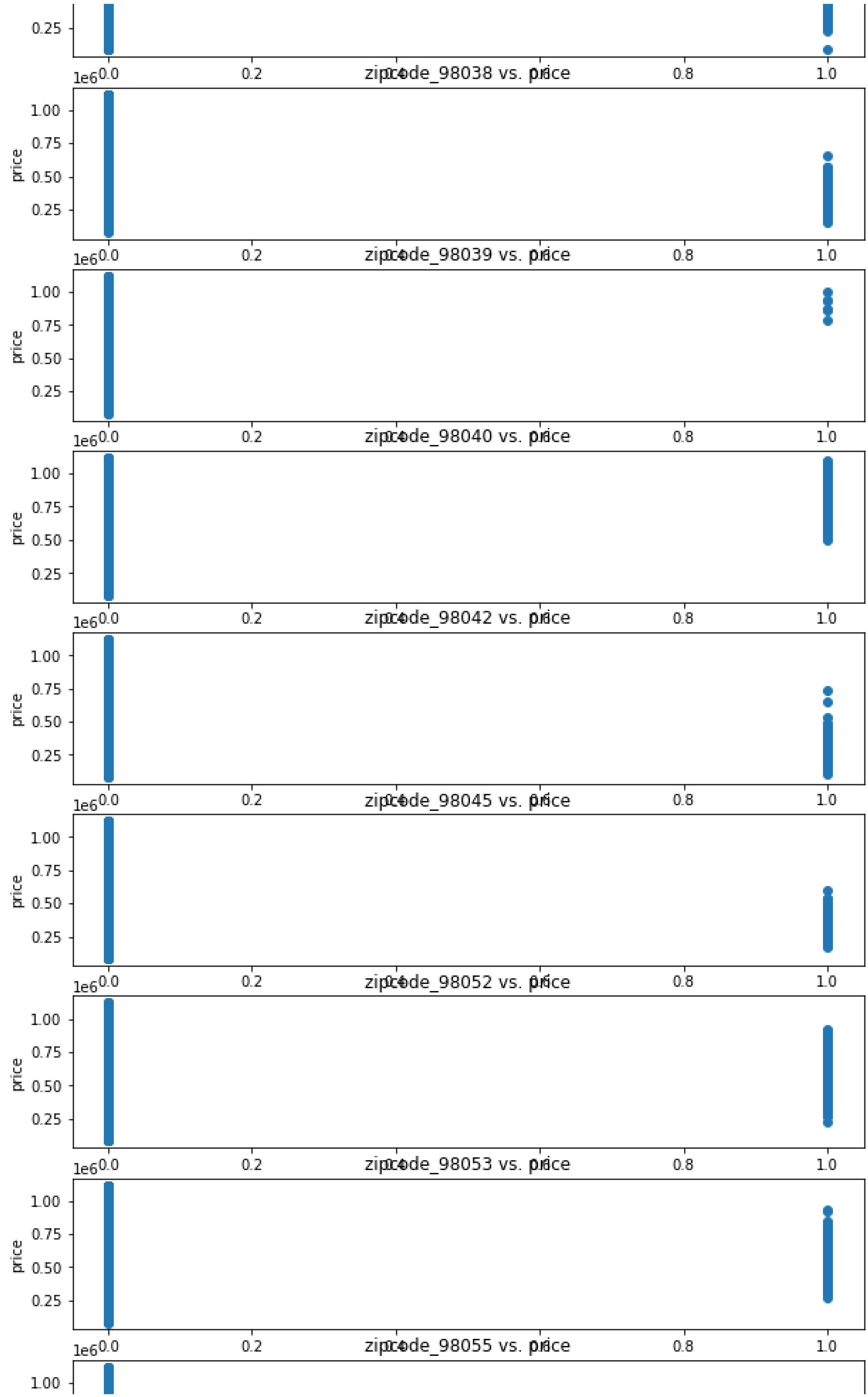


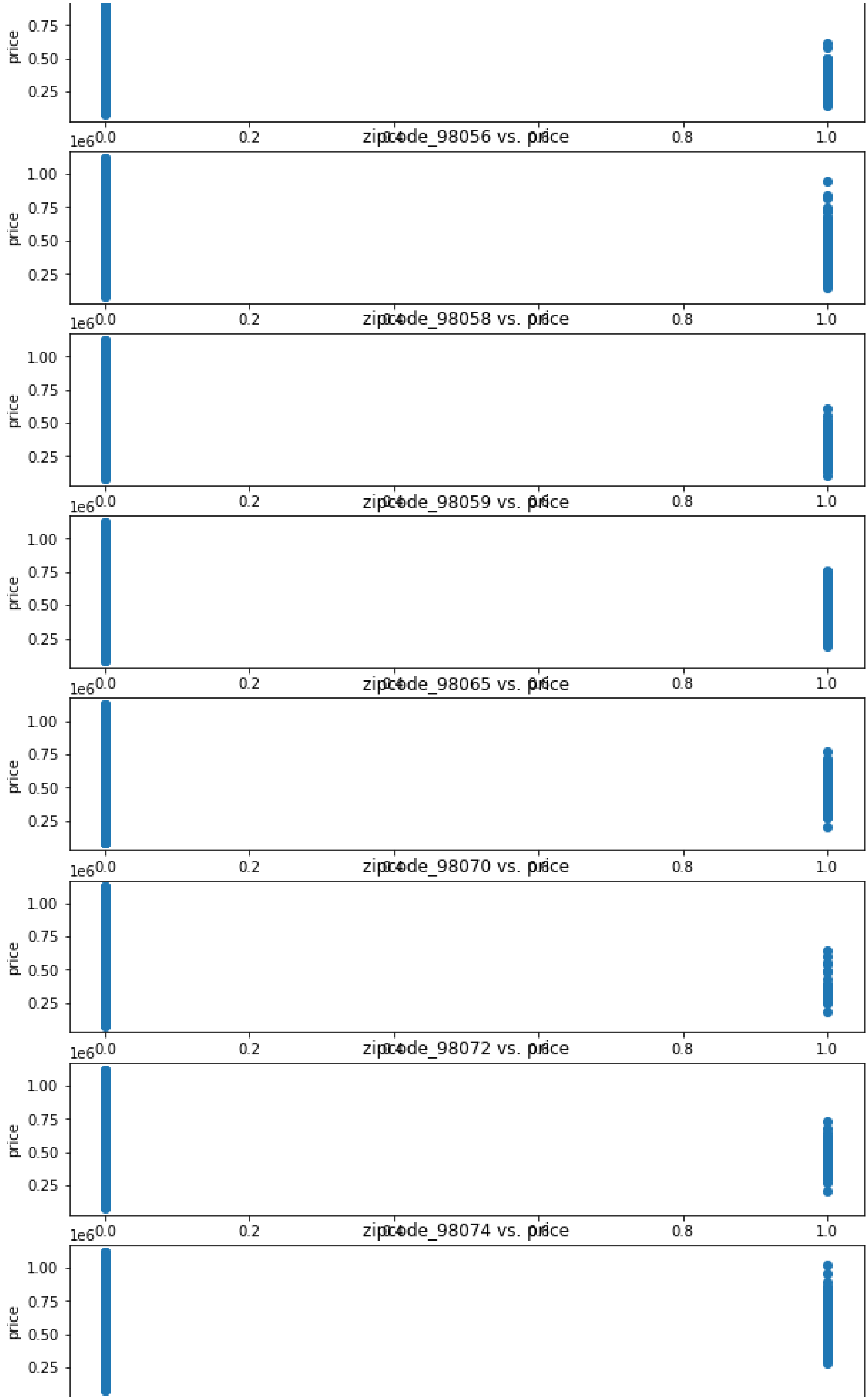

condition vs. price
grade vs. price
sqft_above vs. price
yr_built vs. price
lat vs price
long vs. price
renovated vs. price
price
1e6

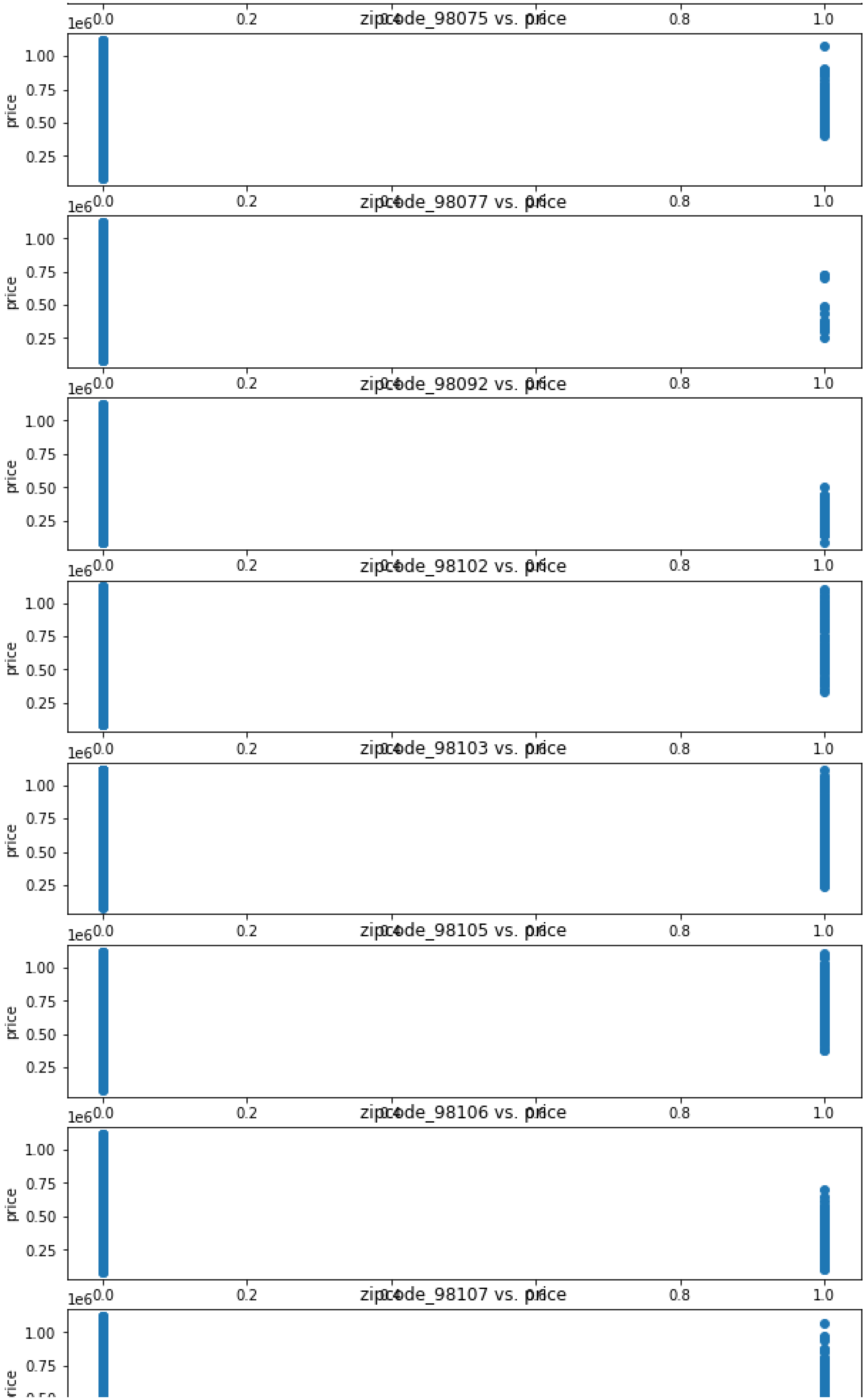

has_basement vs. price
zipcode_98002 vs. price
zipcode_98003 vs. price
zipcode_98004 vs. price
zipcode_98005 vs. price
zipcode_98006 vs. price
zipcode_98007 vs. price
zipcode_98008 vs. price
price
1e6

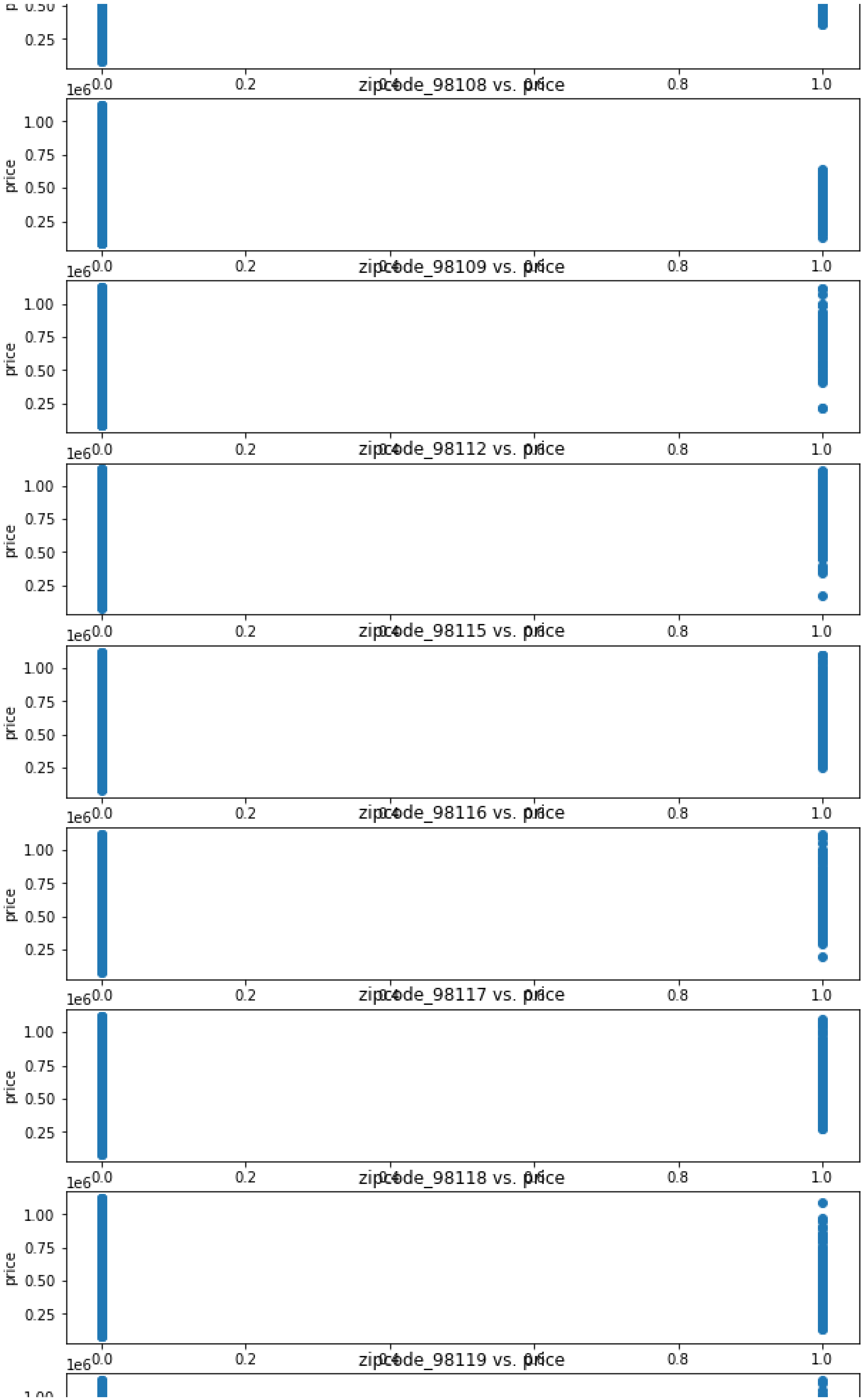

price
zipcode_98010 vs. price
zipcode_98011 vs. price
zipcode_98014 vs. price
zipcode_98019 vs. price
zipcode_98022 vs. price
zipcode_98023 vs. price
zipcode_98024 vs. price
1e6
0.25
0.50
0.75
1.00
0.0
0.2
0.4
0.6
0.8
1.0


zipcode_98027 vs. price
zipcode_98028 vs. price
zipcode_98029 vs. price
zipcode_98030 vs. price
zipcode_98031 vs. price
zipcode_98032 vs. price
zipcode_98033 vs. price
zipcode_98034 vs. price
price
1e6
0.0
0.2
0.4
0.6
0.8
1.0
0.25
0.50
0.75
1.00

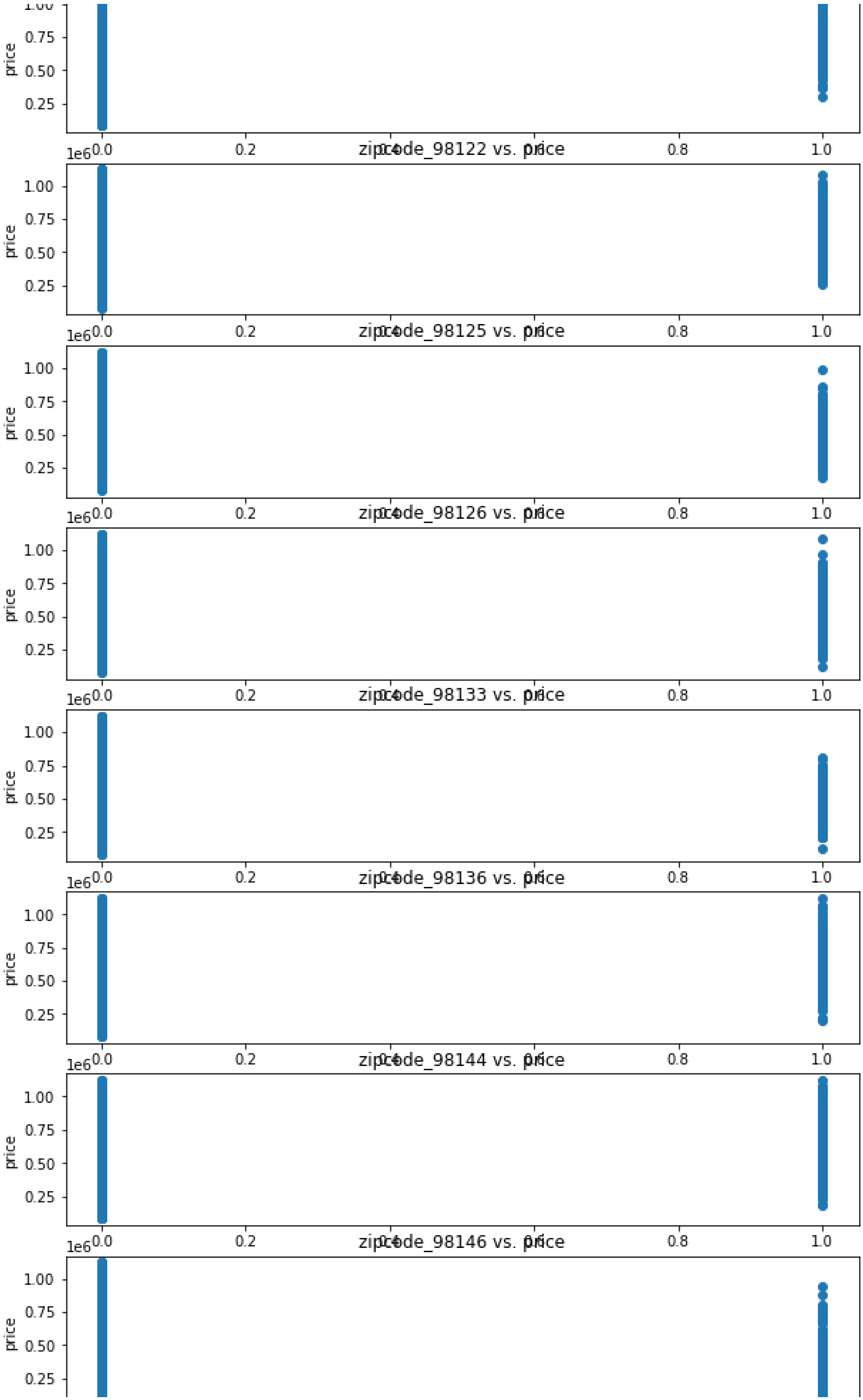

zipcode_98038 vs. price
zipcode_98039 vs. price
zipcode_98040 vs. price
zipcode_98042 vs. price
zipcode_98045 vs. price
zipcode_98052 vs. price
zipcode_98053 vs. price
zipcode_98055 vs. price
price
1e6
0.25
0.50
0.75
1.00
0.0
0.2
0.4
0.6
0.8
1.0

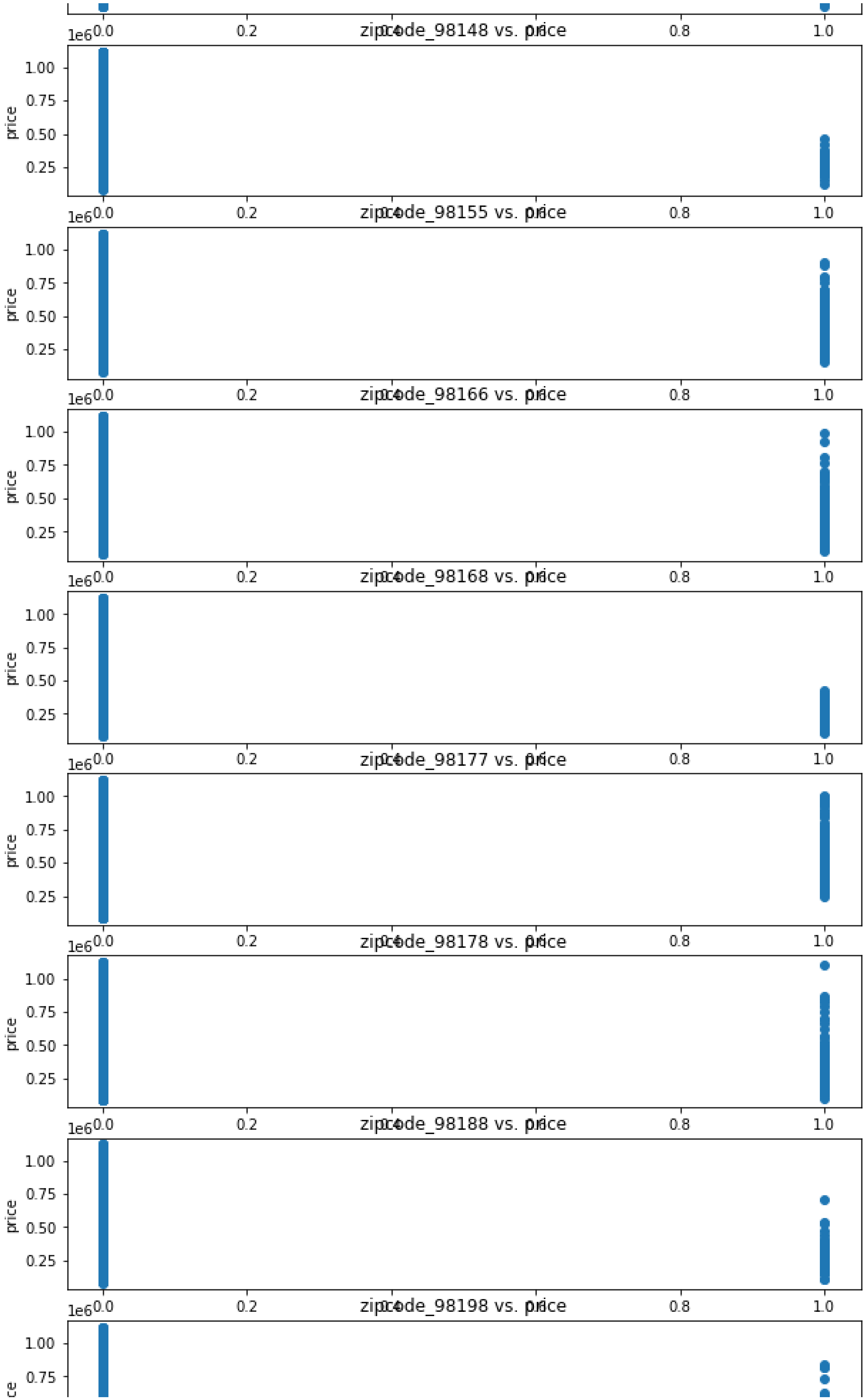

zipcode_98056 vs. price
zipcode_98058 vs. price
zipcode_98059 vs. price
zipcode_98065 vs. price
zipcode_98070 vs. price
zipcode_98072 vs. price
zipcode_98074 vs. price
price
1e6

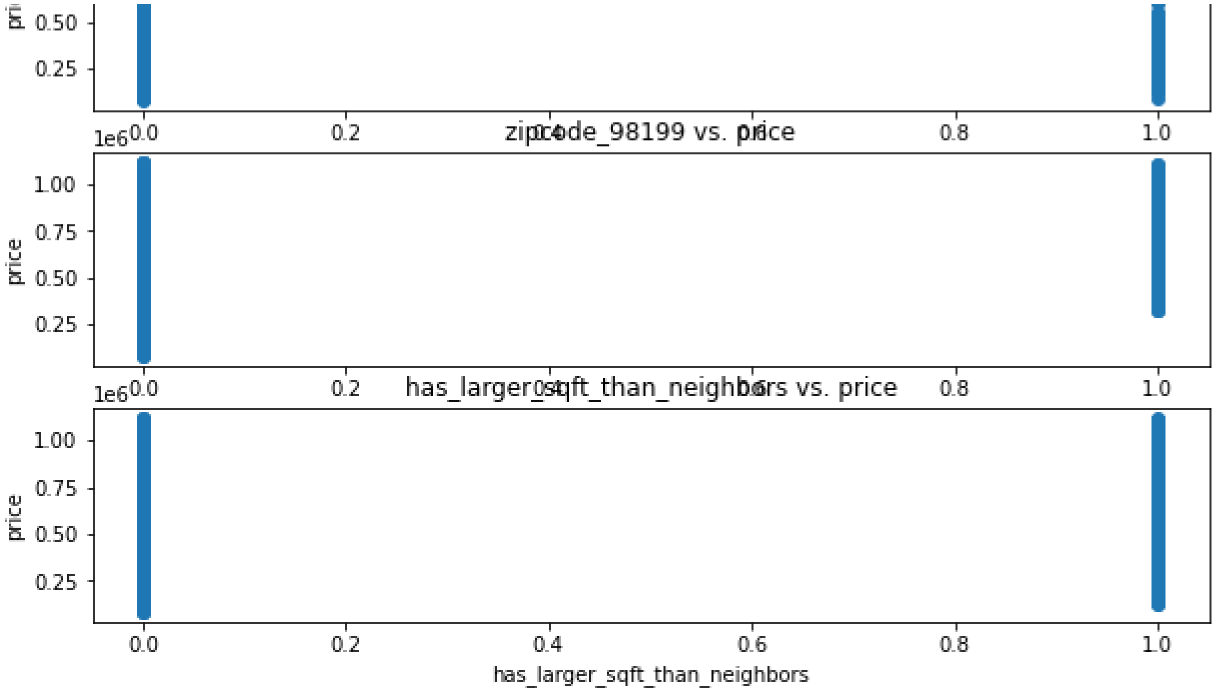

zipcode_98075 vs. price
zipcode_98077 vs. price
zipcode_98092 vs. price
zipcode_98102 vs. price
zipcode_98103 vs. price
zipcode_98105 vs. price
zipcode_98106 vs. price
zipcode_98107 vs. price
price
1e6
0.0
0.2
0.4
0.6
0.8
1.0
0.25
0.50
0.75
1.00



zipcode_98108 vs. price
zipcode_98109 vs. price
zipcode_98112 vs. price
zipcode_98115 vs. price
zipcode_98116 vs. price
zipcode_98117 vs. price
zipcode_98118 vs. price
zipcode_98119 vs. price
price
1e6
0.25
0.50
0.75
1.00
0.0
0.2
0.4
0.6
0.8
1.0


zipcode_98122 vs. price
zipcode_98125 vs. price
zipcode_98126 vs. price
zipcode_98133 vs. price
zipcode_98136 vs. price
zipcode_98144 vs. price
zipcode_98146 vs. price
price
1e6


zipcode_98148 vs. price
zipcode_98155 vs. price
zipcode_98166 vs. price
zipcode_98168 vs. price
zipcode_98177 vs. price
zipcode_98178 vs. price
zipcode_98188 vs. price
zipcode_98198 vs. price
price
1e6
1.00
0.75
0.50
0.25
0.0
0.2
0.4
0.6
0.8
1.0



In [52]:
```
model = model_lin_reg(df=df_ohe)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.820
Model:	OLS	**Adj. R-squared:**	0.819
Method:	Least Squares	**F-statistic:**	890.9
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:18	**Log-Likelihood:**	-2.1045e+05
No. Observations:	16556	**AIC:**	4.211e+05
Df Residuals:	16471	**BIC:**	4.217e+05
Df Model:	84		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-1.459e+06	4.28e+06	-0.341	0.733	-9.85e+06	6.94e+06
bedrooms	2970.0115	1032.190	2.877	0.004	946.808	4993.215
bathrooms	2.089e+04	1587.628	13.158	0.000	1.78e+04	2.4e+04
sqft_lot	1.9798	0.268	7.380	0.000	1.454	2.506
floors	-2.331e+04	1931.424	-12.066	0.000	-2.71e+04	-1.95e+04
waterfront	2.397e+05	1.88e+04	12.747	0.000	2.03e+05	2.77e+05
view	4.034e+04	1180.302	34.180	0.000	3.8e+04	4.27e+04
condition	2.517e+04	1104.208	22.795	0.000	2.3e+04	2.73e+04
grade	4.6e+04	1243.293	37.000	0.000	4.36e+04	4.84e+04

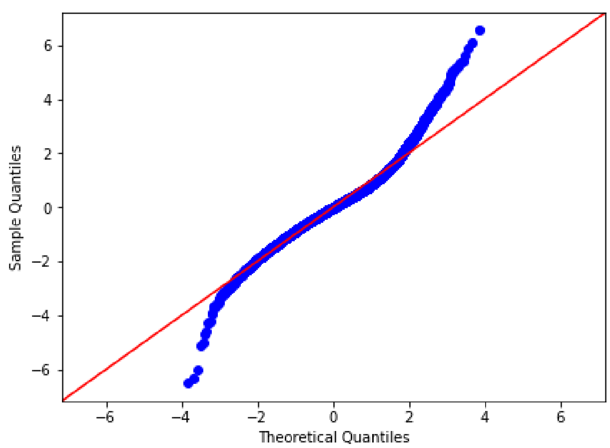
sqft_above	139.7277	2.321	60.207	0.000	135.179	144.277
yr_built	-591.2032	38.117	-15.510	0.000	-665.916	-516.490
lat	-4840.7162	4.02e+04	-0.120	0.904	-8.36e+04	7.39e+04
long	-1.989e+04	3.25e+04	-0.613	0.540	-8.35e+04	4.37e+04
renovated	3.521e+04	3896.615	9.037	0.000	2.76e+04	4.28e+04
has_basement	4.734e+04	1730.499	27.359	0.000	4.4e+04	5.07e+04
zipcode_98002	6751.8751	8013.648	0.843	0.399	-8955.740	2.25e+04
zipcode_98003	-2498.5176	7128.191	-0.351	0.726	-1.65e+04	1.15e+04
zipcode_98004	5.278e+05	1.51e+04	35.017	0.000	4.98e+05	5.57e+05
zipcode_98005	3.347e+05	1.54e+04	21.725	0.000	3.05e+05	3.65e+05
zipcode_98006	2.806e+05	1.28e+04	21.865	0.000	2.55e+05	3.06e+05
zipcode_98007	2.48e+05	1.58e+04	15.711	0.000	2.17e+05	2.79e+05
zipcode_98008	2.408e+05	1.52e+04	15.812	0.000	2.11e+05	2.71e+05
zipcode_98010	8.48e+04	1.61e+04	5.271	0.000	5.33e+04	1.16e+05
zipcode_98011	1.502e+05	1.98e+04	7.596	0.000	1.11e+05	1.89e+05
zipcode_98014	1.139e+05	2.52e+04	4.518	0.000	6.45e+04	1.63e+05
zipcode_98019	1.076e+05	2.2e+04	4.895	0.000	6.45e+04	1.51e+05
zipcode_98022	1843.3567	1.28e+04	0.144	0.886	-2.33e+04	2.69e+04
zipcode_98023	-1.568e+04	6912.313	-2.269	0.023	-2.92e+04	-2136.085
zipcode_98024	1.319e+05	2.4e+04	5.484	0.000	8.47e+04	1.79e+05
zipcode_98027	2.319e+05	1.41e+04	16.451	0.000	2.04e+05	2.59e+05
zipcode_98028	1.39e+05	1.92e+04	7.226	0.000	1.01e+05	1.77e+05
zipcode_98029	2.395e+05	1.54e+04	15.582	0.000	2.09e+05	2.7e+05
zipcode_98030	7538.6621	8010.111	0.941	0.347	-8162.021	2.32e+04
zipcode_98031	1.367e+04	8489.042	1.610	0.107	-2970.898	3.03e+04
zipcode_98032	-7011.8975	9410.308	-0.745	0.456	-2.55e+04	1.14e+04
zipcode_98033	3.175e+05	1.67e+04	19.007	0.000	2.85e+05	3.5e+05
zipcode_98034	1.866e+05	1.78e+04	10.467	0.000	1.52e+05	2.22e+05
zipcode_98038	4.712e+04	1.01e+04	4.652	0.000	2.73e+04	6.7e+04
zipcode_98039	6.561e+05	3.57e+04	18.399	0.000	5.86e+05	7.26e+05
zipcode_98040	4.384e+05	1.34e+04	32.602	0.000	4.12e+05	4.65e+05
zipcode_98042	1.652e+04	8413.643	1.963	0.050	27.240	3.3e+04
zipcode_98045	1.03e+05	1.99e+04	5.185	0.000	6.41e+04	1.42e+05
zipcode_98052	2.576e+05	1.7e+04	15.184	0.000	2.24e+05	2.91e+05
zipcode_98053	2.677e+05	1.94e+04	13.793	0.000	2.3e+05	3.06e+05

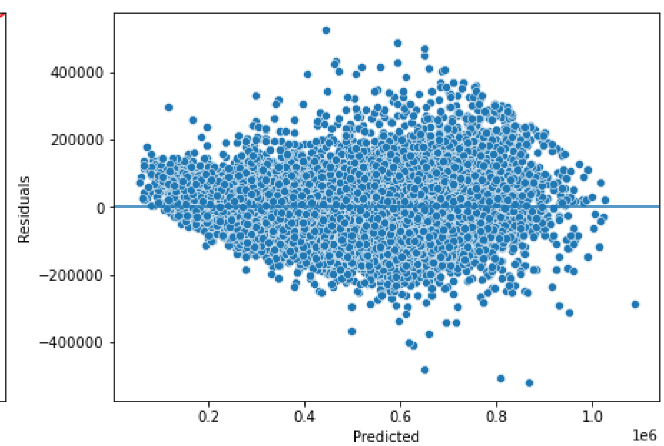
zipcode_98055	4.165e+04	9647.913	4.317	0.000	2.27e+04	6.06e+04
zipcode_98056	1.02e+05	1.09e+04	9.359	0.000	8.06e+04	1.23e+05
zipcode_98058	3.764e+04	9421.517	3.995	0.000	1.92e+04	5.61e+04
zipcode_98059	9.328e+04	1.08e+04	8.662	0.000	7.22e+04	1.14e+05
zipcode_98065	1.486e+05	1.79e+04	8.310	0.000	1.14e+05	1.84e+05
zipcode_98070	4.969e+04	1.84e+04	2.703	0.007	1.37e+04	8.57e+04
zipcode_98072	1.613e+05	2.03e+04	7.967	0.000	1.22e+05	2.01e+05
zipcode_98074	2.159e+05	1.66e+04	13.019	0.000	1.83e+05	2.48e+05
zipcode_98075	2.516e+05	1.69e+04	14.896	0.000	2.19e+05	2.85e+05
zipcode_98077	1.486e+05	2.49e+04	5.980	0.000	9.99e+04	1.97e+05
zipcode_98092	-1.782e+04	7594.534	-2.346	0.019	-3.27e+04	-2929.013
zipcode_98102	4.117e+05	1.67e+04	24.706	0.000	3.79e+05	4.44e+05
zipcode_98103	3.265e+05	1.6e+04	20.422	0.000	2.95e+05	3.58e+05
zipcode_98105	3.773e+05	1.64e+04	22.969	0.000	3.45e+05	4.09e+05
zipcode_98106	1.215e+05	1.15e+04	10.595	0.000	9.91e+04	1.44e+05
zipcode_98107	3.233e+05	1.63e+04	19.820	0.000	2.91e+05	3.55e+05
zipcode_98108	1.229e+05	1.24e+04	9.887	0.000	9.85e+04	1.47e+05
zipcode_98109	4.129e+05	1.68e+04	24.634	0.000	3.8e+05	4.46e+05
zipcode_98112	4.353e+05	1.52e+04	28.561	0.000	4.05e+05	4.65e+05
zipcode_98115	3.232e+05	1.62e+04	19.899	0.000	2.91e+05	3.55e+05
zipcode_98116	2.961e+05	1.3e+04	22.728	0.000	2.71e+05	3.22e+05
zipcode_98117	3.126e+05	1.65e+04	18.945	0.000	2.8e+05	3.45e+05
zipcode_98118	1.693e+05	1.13e+04	15.011	0.000	1.47e+05	1.91e+05
zipcode_98119	4.056e+05	1.59e+04	25.484	0.000	3.74e+05	4.37e+05
zipcode_98122	3.107e+05	1.41e+04	22.080	0.000	2.83e+05	3.38e+05
zipcode_98125	1.964e+05	1.75e+04	11.220	0.000	1.62e+05	2.31e+05
zipcode_98126	1.911e+05	1.18e+04	16.139	0.000	1.68e+05	2.14e+05
zipcode_98133	1.55e+05	1.81e+04	8.553	0.000	1.2e+05	1.91e+05
zipcode_98136	2.547e+05	1.21e+04	21.058	0.000	2.31e+05	2.78e+05
zipcode_98144	2.453e+05	1.31e+04	18.764	0.000	2.2e+05	2.71e+05
zipcode_98146	1.054e+05	1.07e+04	9.876	0.000	8.45e+04	1.26e+05
zipcode_98148	4.602e+04	1.33e+04	3.473	0.001	2e+04	7.2e+04
zipcode_98155	1.424e+05	1.88e+04	7.555	0.000	1.05e+05	1.79e+05
zipcode_98166	8.588e+04	9838.761	8.729	0.000	6.66e+04	1.05e+05
zipcode_98168	5.147e+04	1.03e+04	4.997	0.000	3.13e+04	7.17e+04

zipcode_98177	2.054e+05	1.89e+04	10.855	0.000	1.68e+05	2.43e+05
zipcode_98178	5.611e+04	1.05e+04	5.353	0.000	3.56e+04	7.67e+04
zipcode_98188	3.822e+04	1.04e+04	3.669	0.000	1.78e+04	5.86e+04
zipcode_98198	2.03e+04	7895.317	2.571	0.010	4825.827	3.58e+04
zipcode_98199	3.629e+05	1.57e+04	23.181	0.000	3.32e+05	3.94e+05
has_larger_sqft_than_neighbors	-1.729e+04	1801.693	-9.599	0.000	-2.08e+04	-1.38e+04

Omnibus:	1493.906	**Durbin-Watson:**	1.993
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5130.230
Skew:	0.437	**Prob(JB):**	0.00
Kurtosis:	5.583	**Cond. No.**	5.61e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.61e+07. This might indicate that there are
strong multicollinearity or other numerical problems.





In [53]:
```
model.pvalues[model.pvalues>0.05]
```

Out[53]:
```
Intercept        0.733445
lat              0.904094
long             0.540053
zipcode_98002    0.399494
zipcode_98003    0.725959
zipcode_98022    0.885568
zipcode_98030    0.346645
zipcode_98031    0.107387
zipcode_98032    0.456204
dtype: float64
```

From our p-values, we can see that the latitude and longitude values are insignificant which means that we can go ahead and drop these coefficients.

In [54]:
```
df_ohe.drop(['lat','long'], axis=1, inplace=True)
```

```
In [55]: model = model_lin_reg(df=df_ohe)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.820
Model:	OLS	**Adj. R-squared:**	0.819
Method:	Least Squares	**F-statistic:**	912.7
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:19	**Log-Likelihood:**	-2.1045e+05
No. Observations:	16556	**AIC:**	4.211e+05
Df Residuals:	16473	**BIC:**	4.217e+05
Df Model:	82		
Covariance Type:	nonrobust		

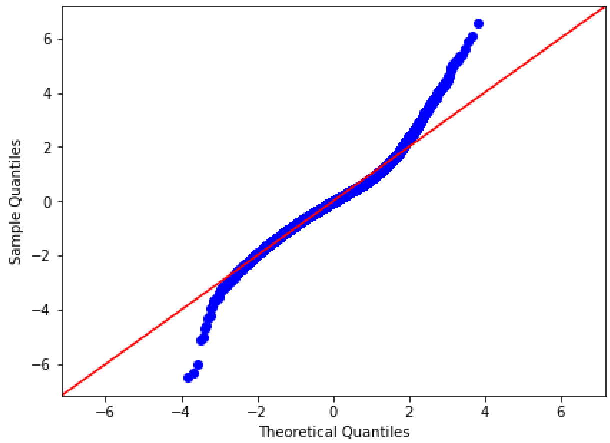
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	7.453e+05	7.43e+04	10.028	0.000	6e+05	8.91e+05
bedrooms	2967.3423	1032.116	2.875	0.004	944.283	4990.401
bathrooms	2.089e+04	1587.508	13.159	0.000	1.78e+04	2.4e+04
sqft_lot	1.9769	0.268	7.371	0.000	1.451	2.503
floors	-2.329e+04	1930.880	-12.061	0.000	-2.71e+04	-1.95e+04
waterfront	2.398e+05	1.88e+04	12.752	0.000	2.03e+05	2.77e+05
view	4.036e+04	1179.978	34.201	0.000	3.8e+04	4.27e+04
condition	2.516e+04	1103.866	22.796	0.000	2.3e+04	2.73e+04
grade	4.602e+04	1242.410	37.043	0.000	4.36e+04	4.85e+04
sqft_above	139.7018	2.320	60.210	0.000	135.154	144.250
yr_built	-591.9980	38.092	-15.541	0.000	-666.663	-517.333
renovated	3.518e+04	3896.042	9.030	0.000	2.75e+04	4.28e+04
has_basement	4.735e+04	1730.358	27.365	0.000	4.4e+04	5.07e+04
zipcode_98002	5621.7555	7796.170	0.721	0.471	-9659.580	2.09e+04
zipcode_98003	-1734.1376	7011.609	-0.247	0.805	-1.55e+04	1.2e+04
zipcode_98004	5.249e+05	8553.918	61.369	0.000	5.08e+05	5.42e+05
zipcode_98005	3.313e+05	9253.837	35.803	0.000	3.13e+05	3.49e+05
zipcode_98006	2.77e+05	6768.602	40.926	0.000	2.64e+05	2.9e+05
zipcode_98007	2.439e+05	9008.742	27.077	0.000	2.26e+05	2.62e+05
zipcode_98008	2.362e+05	7104.393	33.253	0.000	2.22e+05	2.5e+05
zipcode_98010	7.967e+04	1.38e+04	5.785	0.000	5.27e+04	1.07e+05
zipcode_98011	1.466e+05	7965.911	18.408	0.000	1.31e+05	1.62e+05
zipcode_98014	1.033e+05	1.35e+04	7.655	0.000	7.69e+04	1.3e+05

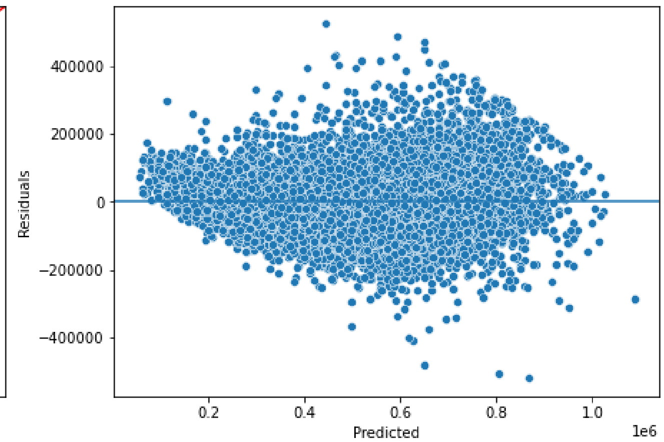
zipcode_98019	9.954e+04	8480.295	11.737	0.000	8.29e+04	1.16e+05
zipcode_98022	-3106.2087	8379.925	-0.371	0.711	-1.95e+04	1.33e+04
zipcode_98023	-1.38e+04	6195.130	-2.227	0.026	-2.59e+04	-1654.491
zipcode_98024	1.233e+05	1.78e+04	6.908	0.000	8.83e+04	1.58e+05
zipcode_98027	2.265e+05	7533.833	30.059	0.000	2.12e+05	2.41e+05
zipcode_98028	1.364e+05	7089.711	19.235	0.000	1.22e+05	1.5e+05
zipcode_98029	2.33e+05	7011.710	33.233	0.000	2.19e+05	2.47e+05
zipcode_98030	5647.8650	7142.783	0.791	0.429	-8352.761	1.96e+04
zipcode_98031	1.16e+04	7060.845	1.643	0.100	-2241.657	2.54e+04
zipcode_98032	-7118.2546	9084.315	-0.784	0.433	-2.49e+04	1.07e+04
zipcode_98033	3.14e+05	6664.977	47.108	0.000	3.01e+05	3.27e+05
zipcode_98034	1.833e+05	6081.409	30.142	0.000	1.71e+05	1.95e+05
zipcode_98038	4.218e+04	6123.375	6.889	0.000	3.02e+04	5.42e+04
zipcode_98039	6.537e+05	3.32e+04	19.684	0.000	5.89e+05	7.19e+05
zipcode_98040	4.362e+05	8660.795	50.370	0.000	4.19e+05	4.53e+05
zipcode_98042	1.316e+04	6237.511	2.111	0.035	938.149	2.54e+04
zipcode_98045	9.222e+04	8411.472	10.963	0.000	7.57e+04	1.09e+05
zipcode_98052	2.528e+05	6211.997	40.696	0.000	2.41e+05	2.65e+05
zipcode_98053	2.61e+05	7611.304	34.286	0.000	2.46e+05	2.76e+05
zipcode_98055	3.957e+04	7202.509	5.494	0.000	2.55e+04	5.37e+04
zipcode_98056	9.924e+04	6610.285	15.013	0.000	8.63e+04	1.12e+05
zipcode_98058	3.462e+04	6422.863	5.391	0.000	2.2e+04	4.72e+04
zipcode_98059	8.987e+04	6561.505	13.696	0.000	7.7e+04	1.03e+05
zipcode_98065	1.395e+05	7348.861	18.986	0.000	1.25e+05	1.54e+05
zipcode_98070	5.291e+04	1.7e+04	3.114	0.002	1.96e+04	8.62e+04
zipcode_98072	1.567e+05	8384.514	18.688	0.000	1.4e+05	1.73e+05
zipcode_98074	2.098e+05	6994.896	29.998	0.000	1.96e+05	2.24e+05
zipcode_98075	2.455e+05	9339.484	26.284	0.000	2.27e+05	2.64e+05
zipcode_98077	1.426e+05	1.6e+04	8.928	0.000	1.11e+05	1.74e+05
zipcode_98092	-1.948e+04	7070.251	-2.755	0.006	-3.33e+04	-5617.113
zipcode_98102	4.111e+05	1.05e+04	39.314	0.000	3.91e+05	4.32e+05
zipcode_98103	3.262e+05	6333.106	51.503	0.000	3.14e+05	3.39e+05
zipcode_98105	3.761e+05	8236.550	45.658	0.000	3.6e+05	3.92e+05
zipcode_98106	1.221e+05	6771.130	18.040	0.000	1.09e+05	1.35e+05
zipcode_98107	3.236e+05	7348.915	44.036	0.000	3.09e+05	3.38e+05

zipcode_98108	1.224e+05	7909.632	15.476	0.000	1.07e+05	1.38e+05
zipcode_98109	4.129e+05	1.05e+04	39.385	0.000	3.92e+05	4.33e+05
zipcode_98112	4.343e+05	8413.952	51.614	0.000	4.18e+05	4.51e+05
zipcode_98115	3.22e+05	6227.641	51.704	0.000	3.1e+05	3.34e+05
zipcode_98116	2.972e+05	7015.178	42.366	0.000	2.83e+05	3.11e+05
zipcode_98117	3.129e+05	6336.595	49.383	0.000	3e+05	3.25e+05
zipcode_98118	1.682e+05	6318.881	26.620	0.000	1.56e+05	1.81e+05
zipcode_98119	4.059e+05	8612.640	47.123	0.000	3.89e+05	4.23e+05
zipcode_98122	3.098e+05	7390.263	41.914	0.000	2.95e+05	3.24e+05
zipcode_98125	1.95e+05	6465.387	30.163	0.000	1.82e+05	2.08e+05
zipcode_98126	1.92e+05	6757.688	28.408	0.000	1.79e+05	2.05e+05
zipcode_98133	1.544e+05	6158.422	25.075	0.000	1.42e+05	1.66e+05
zipcode_98136	2.559e+05	7318.534	34.961	0.000	2.42e+05	2.7e+05
zipcode_98144	2.445e+05	7016.514	34.846	0.000	2.31e+05	2.58e+05
zipcode_98146	1.062e+05	7089.385	14.980	0.000	9.23e+04	1.2e+05
zipcode_98148	4.658e+04	1.22e+04	3.826	0.000	2.27e+04	7.04e+04
zipcode_98155	1.409e+05	6318.872	22.302	0.000	1.29e+05	1.53e+05
zipcode_98166	8.672e+04	7622.316	11.378	0.000	7.18e+04	1.02e+05
zipcode_98168	5.135e+04	7361.707	6.975	0.000	3.69e+04	6.58e+04
zipcode_98177	2.053e+05	7624.774	26.920	0.000	1.9e+05	2.2e+05
zipcode_98178	5.47e+04	7156.946	7.643	0.000	4.07e+04	6.87e+04
zipcode_98188	3.776e+04	8882.117	4.251	0.000	2.03e+04	5.52e+04
zipcode_98198	2.079e+04	7116.150	2.922	0.003	6844.626	3.47e+04
zipcode_98199	3.638e+05	7274.106	50.009	0.000	3.5e+05	3.78e+05
has_larger_sqft_than_neighbors	-1.728e+04	1801.049	-9.595	0.000	-2.08e+04	-1.38e+04

Omnibus:	1494.930	**Durbin-Watson:**	1.993
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5136.551
Skew:	0.438	**Prob(JB):**	0.00
Kurtosis:	5.585	**Cond. No.**	9.83e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.83e+05. This might indicate that there are
strong multicollinearity or other numerical problems.





Thanks to the IQR outlier removal process, our residuals seem to now be better fitting into the normality and homoscedasticity assumptions.

In [56]:
```
df_ohe
```

Out[56]:

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_abov
0	221900.0	3	1.00	5650	1.0	0.0	0.0	3	7	118
1	538000.0	3	2.25	7242	2.0	0.0	0.0	3	7	217
2	180000.0	2	1.00	10000	1.0	0.0	0.0	3	6	77
3	604000.0	4	3.00	5000	1.0	0.0	0.0	5	7	105
4	510000.0	3	2.00	8080	1.0	0.0	0.0	3	8	168
...	...	...	...	...	...	...	...	...	...	.
21592	360000.0	3	2.50	1131	3.0	0.0	0.0	3	8	153
21593	400000.0	4	2.50	5813	2.0	0.0	0.0	3	8	231
21594	402101.0	2	0.75	1350	2.0	0.0	0.0	3	7	102
21595	400000.0	3	2.50	2388	2.0	0.0	0.0	3	8	160
21596	325000.0	2	0.75	1076	2.0	0.0	0.0	3	7	102

16556 rows × 83 columns

Since we removed outliers from all numeric columns including our target 'price', we are left with 16556 rows compared to the 21597 we started off with. It makes more sense to remove outliers based on the prices of the homes rather than removing outliers in every single column. This has a potential upside of allowing for there to be more data points and therefore a more accurate model overall.

Removing Outliers Based on Price Only

In [57]:
```
df_IQR_price = df_IQR_price[find_outliers_IQR(df_IQR_price['price'])==False]
df_IQR_price
```

Out[57]:

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_abov

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	sqft_abov
0	221900.0	3	1.00	5650	1.0	0.0	0.0	3	7	118
1	538000.0	3	2.25	7242	2.0	0.0	0.0	3	7	217
2	180000.0	2	1.00	10000	1.0	0.0	0.0	3	6	77
3	604000.0	4	3.00	5000	1.0	0.0	0.0	5	7	105
4	510000.0	3	2.00	8080	1.0	0.0	0.0	3	8	168
...	...	...	...	...	...	...	...	...	...	
21592	360000.0	3	2.50	1131	3.0	0.0	0.0	3	8	153
21593	400000.0	4	2.50	5813	2.0	0.0	0.0	3	8	231
21594	402101.0	2	0.75	1350	2.0	0.0	0.0	3	7	102
21595	400000.0	3	2.50	2388	2.0	0.0	0.0	3	8	160
21596	325000.0	2	0.75	1076	2.0	0.0	0.0	3	7	102

20439 rows × 85 columns

As seen above, we are left with approximately 4,000 more data points when we only remove outliers based on price. We still should take a look at the model and whether the residuals have adjusted similarly to the prior model.

In [58]:
```
model = model_lin_reg(df=df_IQR_price)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.826
Model:	OLS	**Adj. R-squared:**	0.826
Method:	Least Squares	**F-statistic:**	1152.
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:20	**Log-Likelihood:**	-2.6138e+05
No. Observations:	20439	**AIC:**	5.229e+05
Df Residuals:	20354	**BIC:**	5.236e+05
Df Model:	84		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.461e+07	3.39e+06	-4.308	0.000	-2.13e+07	-7.96e+06
bedrooms	-244.3677	850.207	-0.287	0.774	-1910.842	1422.106
bathrooms	1.95e+04	1472.657	13.242	0.000	1.66e+04	2.24e+04
sqft_lot	0.3002	0.017	17.975	0.000	0.268	0.333
floors	-2.479e+04	1761.730	-14.073	0.000	-2.82e+04	-2.13e+04

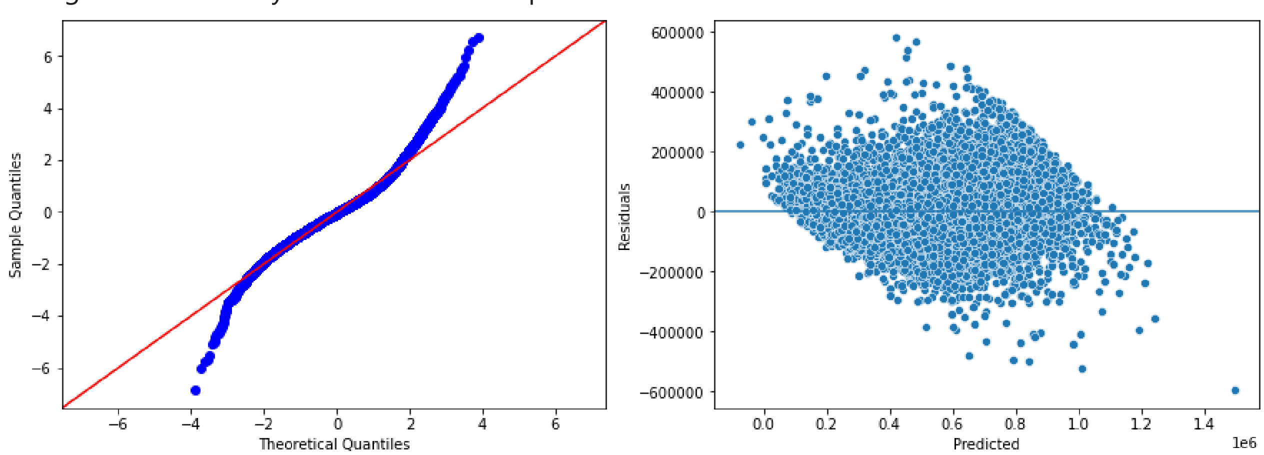
waterfront	1.501e+05	1.31e+04	11.488	0.000	1.24e+05	1.76e+05
view	3.731e+04	1052.530	35.448	0.000	3.52e+04	3.94e+04
condition	2.5e+04	1068.482	23.400	0.000	2.29e+04	2.71e+04
grade	4.831e+04	1030.415	46.879	0.000	4.63e+04	5.03e+04
sqft_above	131.7060	1.883	69.954	0.000	128.016	135.396
yr_built	-572.7285	36.232	-15.807	0.000	-643.747	-501.710
lat	1.545e+05	3.51e+04	4.402	0.000	8.57e+04	2.23e+05
long	-6.569e+04	2.51e+04	-2.621	0.009	-1.15e+05	-1.66e+04
renovated	3.36e+04	3726.684	9.017	0.000	2.63e+04	4.09e+04
has_basement	4.727e+04	1670.349	28.300	0.000	4.4e+04	5.05e+04
zipcode_98002	6083.3919	7819.865	0.778	0.437	-9244.173	2.14e+04
zipcode_98003	-1.129e+04	6991.183	-1.615	0.106	-2.5e+04	2411.862
zipcode_98004	4.749e+05	1.37e+04	34.573	0.000	4.48e+05	5.02e+05
zipcode_98005	2.94e+05	1.39e+04	21.181	0.000	2.67e+05	3.21e+05
zipcode_98006	2.484e+05	1.14e+04	21.769	0.000	2.26e+05	2.71e+05
zipcode_98007	2.177e+05	1.42e+04	15.285	0.000	1.9e+05	2.46e+05
zipcode_98008	2.006e+05	1.36e+04	14.753	0.000	1.74e+05	2.27e+05
zipcode_98010	9.965e+04	1.2e+04	8.310	0.000	7.61e+04	1.23e+05
zipcode_98011	8.511e+04	1.76e+04	4.825	0.000	5.05e+04	1.2e+05
zipcode_98014	8.339e+04	1.94e+04	4.297	0.000	4.54e+04	1.21e+05
zipcode_98019	5.97e+04	1.91e+04	3.127	0.002	2.23e+04	9.71e+04
zipcode_98022	2.401e+04	1.05e+04	2.291	0.022	3470.958	4.45e+04
zipcode_98023	-2.911e+04	6450.507	-4.513	0.000	-4.18e+04	-1.65e+04
zipcode_98024	1.335e+05	1.72e+04	7.770	0.000	9.98e+04	1.67e+05
zipcode_98027	1.777e+05	1.16e+04	15.365	0.000	1.55e+05	2e+05
zipcode_98028	7.243e+04	1.72e+04	4.221	0.000	3.88e+04	1.06e+05
zipcode_98029	2.089e+05	1.32e+04	15.786	0.000	1.83e+05	2.35e+05
zipcode_98030	1427.2186	7704.548	0.185	0.853	-1.37e+04	1.65e+04
zipcode_98031	2295.0973	8044.248	0.285	0.775	-1.35e+04	1.81e+04
zipcode_98032	-1.8e+04	9302.376	-1.935	0.053	-3.62e+04	235.348
zipcode_98033	2.702e+05	1.48e+04	18.204	0.000	2.41e+05	2.99e+05
zipcode_98034	1.25e+05	1.58e+04	7.906	0.000	9.4e+04	1.56e+05
zipcode_98038	5e+04	8718.347	5.735	0.000	3.29e+04	6.71e+04
zipcode_98039	6.029e+05	3.75e+04	16.072	0.000	5.29e+05	6.76e+05
zipcode_98040	3.904e+05	1.22e+04	32.021	0.000	3.67e+05	4.14e+05

zipcode_98042	1.043e+04	7403.975	1.409	0.159	-4081.403	2.49e+04
zipcode_98045	1.112e+05	1.62e+04	6.866	0.000	7.94e+04	1.43e+05
zipcode_98052	2.092e+05	1.5e+04	13.929	0.000	1.8e+05	2.39e+05
zipcode_98053	2.011e+05	1.61e+04	12.466	0.000	1.7e+05	2.33e+05
zipcode_98055	2.19e+04	8989.534	2.436	0.015	4281.690	3.95e+04
zipcode_98056	7.814e+04	9814.522	7.961	0.000	5.89e+04	9.74e+04
zipcode_98058	2.414e+04	8506.076	2.838	0.005	7466.703	4.08e+04
zipcode_98059	8.623e+04	9615.356	8.968	0.000	6.74e+04	1.05e+05
zipcode_98065	1.264e+05	1.49e+04	8.473	0.000	9.72e+04	1.56e+05
zipcode_98070	7.2e+04	1.14e+04	6.334	0.000	4.97e+04	9.43e+04
zipcode_98072	1.227e+05	1.76e+04	6.978	0.000	8.82e+04	1.57e+05
zipcode_98074	1.88e+05	1.42e+04	13.231	0.000	1.6e+05	2.16e+05
zipcode_98075	2.145e+05	1.37e+04	15.671	0.000	1.88e+05	2.41e+05
zipcode_98077	1.269e+05	1.83e+04	6.929	0.000	9.1e+04	1.63e+05
zipcode_98092	-7663.2632	7003.594	-1.094	0.274	-2.14e+04	6064.345
zipcode_98102	3.42e+05	1.55e+04	22.013	0.000	3.12e+05	3.72e+05
zipcode_98103	2.507e+05	1.43e+04	17.591	0.000	2.23e+05	2.79e+05
zipcode_98105	3.089e+05	1.49e+04	20.695	0.000	2.8e+05	3.38e+05
zipcode_98106	7.308e+04	1.05e+04	6.969	0.000	5.25e+04	9.36e+04
zipcode_98107	2.481e+05	1.47e+04	16.910	0.000	2.19e+05	2.77e+05
zipcode_98108	7.486e+04	1.16e+04	6.476	0.000	5.22e+04	9.75e+04
zipcode_98109	3.432e+05	1.56e+04	21.980	0.000	3.13e+05	3.74e+05
zipcode_98112	3.687e+05	1.4e+04	26.416	0.000	3.41e+05	3.96e+05
zipcode_98115	2.535e+05	1.45e+04	17.488	0.000	2.25e+05	2.82e+05
zipcode_98116	2.366e+05	1.18e+04	20.095	0.000	2.13e+05	2.6e+05
zipcode_98117	2.387e+05	1.47e+04	16.278	0.000	2.1e+05	2.67e+05
zipcode_98118	1.236e+05	1.02e+04	12.064	0.000	1.04e+05	1.44e+05
zipcode_98119	3.321e+05	1.45e+04	22.835	0.000	3.04e+05	3.61e+05
zipcode_98122	2.458e+05	1.28e+04	19.265	0.000	2.21e+05	2.71e+05
zipcode_98125	1.291e+05	1.56e+04	8.261	0.000	9.85e+04	1.6e+05
zipcode_98126	1.385e+05	1.08e+04	12.873	0.000	1.17e+05	1.6e+05
zipcode_98133	7.66e+04	1.61e+04	4.744	0.000	4.5e+04	1.08e+05
zipcode_98136	2.047e+05	1.11e+04	18.520	0.000	1.83e+05	2.26e+05
zipcode_98144	1.918e+05	1.19e+04	16.157	0.000	1.68e+05	2.15e+05
zipcode_98146	7.139e+04	9838.756	7.256	0.000	5.21e+04	9.07e+04

zipcode_98148	3.006e+04	1.32e+04	2.273	0.023	4139.841	5.6e+04
zipcode_98155	6.584e+04	1.68e+04	3.919	0.000	3.29e+04	9.88e+04
zipcode_98166	7.007e+04	9010.639	7.777	0.000	5.24e+04	8.77e+04
zipcode_98168	1.879e+04	9482.888	1.981	0.048	201.759	3.74e+04
zipcode_98177	1.401e+05	1.69e+04	8.267	0.000	1.07e+05	1.73e+05
zipcode_98178	2.552e+04	9804.820	2.602	0.009	6298.076	4.47e+04
zipcode_98188	1.038e+04	9994.135	1.039	0.299	-9209.027	3e+04
zipcode_98198	5223.4014	7582.058	0.689	0.491	-9638.043	2.01e+04
zipcode_98199	2.926e+05	1.4e+04	20.843	0.000	2.65e+05	3.2e+05
has_larger_sqft_than_neighbors	-1.718e+04	1719.318	-9.993	0.000	-2.06e+04	-1.38e+04

Omnibus:	1997.283	**Durbin-Watson:**	1.988
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	7373.063
Skew:	0.457	**Prob(JB):**	0.00
Kurtosis:	5.797	**Cond. No.**	2.38e+08

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.38e+08. This might indicate that there are
strong multicollinearity or other numerical problems.



Our model has a higher R-sqaured value at 0.826 compared to the prior model at 0.820 and has more data points. We will keep iterating on this model instead of using the prior one due to this reason.

In [59]:
```
model.pvalues[model.pvalues>0.05]
```

Out[59]:
```
bedrooms         0.773793
zipcode_98002    0.436613
zipcode_98003    0.106306
zipcode_98030    0.853040
zipcode_98031    0.775410
```

```
zipcode_98032    0.053031
zipcode_98042    0.158899
zipcode_98092    0.273885
zipcode_98188    0.298986
zipcode_98198    0.490884
dtype: float64
```

Interestingly, we are seeing that bedrooms are no longer a significant coefficient, and therefore not a significant parameter to define the sales price of a home based on our model. We can keep the 'bedrooms' parameter in for the time being to see if anything changes when we scale the model to compare our parameters' effects against each other. A quick note compared to the previous model we had is that 'lat' and 'long' are significant in this model so we are keeping them in.

Scaling

In [60]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

In [61]:
```
df_IQR_price.columns
```

Out[61]:
```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors', 'waterfront',
       'view', 'condition', 'grade', 'sqft_above', 'yr_built', 'lat', 'long',
       'renovated', 'has_basement', 'zipcode_98002', 'zipcode_98003',
       'zipcode_98004', 'zipcode_98005', 'zipcode_98006', 'zipcode_98007',
       'zipcode_98008', 'zipcode_98010', 'zipcode_98011', 'zipcode_98014',
       'zipcode_98019', 'zipcode_98022', 'zipcode_98023', 'zipcode_98024',
       'zipcode_98027', 'zipcode_98028', 'zipcode_98029', 'zipcode_98030',
       'zipcode_98031', 'zipcode_98032', 'zipcode_98033', 'zipcode_98034',
       'zipcode_98038', 'zipcode_98039', 'zipcode_98040', 'zipcode_98042',
       'zipcode_98045', 'zipcode_98052', 'zipcode_98053', 'zipcode_98055',
       'zipcode_98056', 'zipcode_98058', 'zipcode_98059', 'zipcode_98065',
       'zipcode_98070', 'zipcode_98072', 'zipcode_98074', 'zipcode_98075',
       'zipcode_98077', 'zipcode_98092', 'zipcode_98102', 'zipcode_98103',
       'zipcode_98105', 'zipcode_98106', 'zipcode_98107', 'zipcode_98108',
       'zipcode_98109', 'zipcode_98112', 'zipcode_98115', 'zipcode_98116',
       'zipcode_98117', 'zipcode_98118', 'zipcode_98119', 'zipcode_98122',
       'zipcode_98125', 'zipcode_98126', 'zipcode_98133', 'zipcode_98136',
       'zipcode_98144', 'zipcode_98146', 'zipcode_98148', 'zipcode_98155',
       'zipcode_98166', 'zipcode_98168', 'zipcode_98177', 'zipcode_98178',
       'zipcode_98188', 'zipcode_98198', 'zipcode_98199',
       'has_larger_sqft_than_neighbors'],
      dtype='object')
```

In [62]:
```
numeric_cols = [col for col in df_IQR_price.columns if (col.startswith('zipcode')==Fals
```

In [63]:
```
numeric_cols
```

Out[63]:
```
['bedrooms',
 'bathrooms',
 'sqft_lot',
 'floors',
 'waterfront',
 'view',
 'condition',
 'grade',
 'sqft_above',
 'yr_built',
 'lat',
 'long',
 'renovated']
```

In [64]:
```
df_scaled = df_IQR_price.copy()
df_scaled[numeric_cols] = scaler.fit_transform(df_scaled[numeric_cols])
df_scaled.head()
```

Out[64]:

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	view	condition	grade	s
0	221900.0	-0.364398	-1.481158	-0.223340	-0.886223	-0.050015	-0.268494	-0.628492	-0.512196	
1	538000.0	-0.364398	0.278964	-0.183549	0.977307	-0.050015	-0.268494	-0.628492	-0.512196	
2	180000.0	-1.465513	-1.481158	-0.114614	-0.886223	-0.050015	-0.268494	-0.628492	-1.477415	
3	604000.0	0.736716	1.335036	-0.239586	-0.886223	-0.050015	-0.268494	2.462772	-0.512196	
4	510000.0	-0.364398	-0.073061	-0.162603	-0.886223	-0.050015	-0.268494	-0.628492	0.453023	

5 rows × 85 columns

In [65]:
```
model = model_lin_reg(df=df_scaled)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.826
Model:	OLS	**Adj. R-squared:**	0.826
Method:	Least Squares	**F-statistic:**	1152.
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:21	**Log-Likelihood:**	-2.6138e+05
No. Observations:	20439	**AIC:**	5.229e+05
Df Residuals:	20354	**BIC:**	5.236e+05
Df Model:	84		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.302e+05	1e+04	32.869	0.000	3.11e+05	3.5e+05
bedrooms	-221.9276	772.133	-0.287	0.774	-1735.370	1291.515
bathrooms	1.385e+04	1045.849	13.242	0.000	1.18e+04	1.59e+04
sqft_lot	1.201e+04	668.299	17.975	0.000	1.07e+04	1.33e+04
floors	-1.33e+04	945.372	-14.073	0.000	-1.52e+04	-1.15e+04
waterfront	7486.4714	651.680	11.488	0.000	6209.126	8763.817
view	2.384e+04	672.630	35.448	0.000	2.25e+04	2.52e+04
condition	1.618e+04	691.292	23.400	0.000	1.48e+04	1.75e+04
grade	5.005e+04	1067.545	46.879	0.000	4.8e+04	5.21e+04
sqft_above	9.574e+04	1368.559	69.954	0.000	9.31e+04	9.84e+04
yr_built	-1.67e+04	1056.472	-15.807	0.000	-1.88e+04	-1.46e+04
lat	2.18e+04	4951.117	4.402	0.000	1.21e+04	3.15e+04

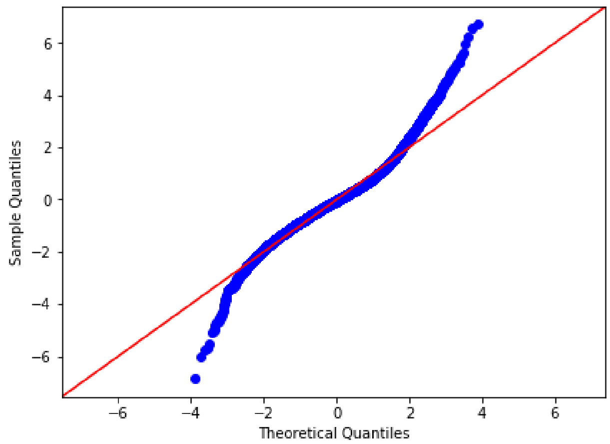
long	-9349.0429	3567.175	-2.621	0.009	-1.63e+04	-2357.092
renovated	5785.7697	641.636	9.017	0.000	4528.111	7043.429
has_basement	4.727e+04	1670.349	28.300	0.000	4.4e+04	5.05e+04
zipcode_98002	6083.3919	7819.865	0.778	0.437	-9244.173	2.14e+04
zipcode_98003	-1.129e+04	6991.183	-1.615	0.106	-2.5e+04	2411.862
zipcode_98004	4.749e+05	1.37e+04	34.573	0.000	4.48e+05	5.02e+05
zipcode_98005	2.94e+05	1.39e+04	21.181	0.000	2.67e+05	3.21e+05
zipcode_98006	2.484e+05	1.14e+04	21.769	0.000	2.26e+05	2.71e+05
zipcode_98007	2.177e+05	1.42e+04	15.285	0.000	1.9e+05	2.46e+05
zipcode_98008	2.006e+05	1.36e+04	14.753	0.000	1.74e+05	2.27e+05
zipcode_98010	9.965e+04	1.2e+04	8.310	0.000	7.61e+04	1.23e+05
zipcode_98011	8.511e+04	1.76e+04	4.825	0.000	5.05e+04	1.2e+05
zipcode_98014	8.339e+04	1.94e+04	4.297	0.000	4.54e+04	1.21e+05
zipcode_98019	5.97e+04	1.91e+04	3.127	0.002	2.23e+04	9.71e+04
zipcode_98022	2.401e+04	1.05e+04	2.291	0.022	3470.958	4.45e+04
zipcode_98023	-2.911e+04	6450.507	-4.513	0.000	-4.18e+04	-1.65e+04
zipcode_98024	1.335e+05	1.72e+04	7.770	0.000	9.98e+04	1.67e+05
zipcode_98027	1.777e+05	1.16e+04	15.365	0.000	1.55e+05	2e+05
zipcode_98028	7.243e+04	1.72e+04	4.221	0.000	3.88e+04	1.06e+05
zipcode_98029	2.089e+05	1.32e+04	15.786	0.000	1.83e+05	2.35e+05
zipcode_98030	1427.2186	7704.548	0.185	0.853	-1.37e+04	1.65e+04
zipcode_98031	2295.0973	8044.248	0.285	0.775	-1.35e+04	1.81e+04
zipcode_98032	-1.8e+04	9302.376	-1.935	0.053	-3.62e+04	235.348
zipcode_98033	2.702e+05	1.48e+04	18.204	0.000	2.41e+05	2.99e+05
zipcode_98034	1.25e+05	1.58e+04	7.906	0.000	9.4e+04	1.56e+05
zipcode_98038	5e+04	8718.347	5.735	0.000	3.29e+04	6.71e+04
zipcode_98039	6.029e+05	3.75e+04	16.072	0.000	5.29e+05	6.76e+05
zipcode_98040	3.904e+05	1.22e+04	32.021	0.000	3.67e+05	4.14e+05
zipcode_98042	1.043e+04	7403.975	1.409	0.159	-4081.403	2.49e+04
zipcode_98045	1.112e+05	1.62e+04	6.866	0.000	7.94e+04	1.43e+05
zipcode_98052	2.092e+05	1.5e+04	13.929	0.000	1.8e+05	2.39e+05
zipcode_98053	2.011e+05	1.61e+04	12.466	0.000	1.7e+05	2.33e+05
zipcode_98055	2.19e+04	8989.534	2.436	0.015	4281.690	3.95e+04
zipcode_98056	7.814e+04	9814.522	7.961	0.000	5.89e+04	9.74e+04
zipcode_98058	2.414e+04	8506.076	2.838	0.005	7466.703	4.08e+04

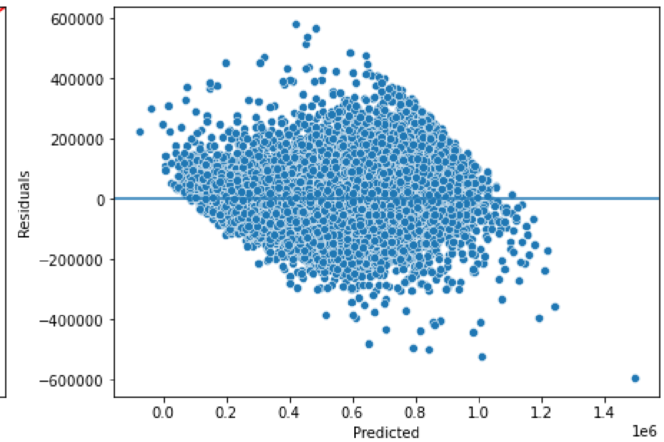
zipcode_98059	8.623e+04	9615.356	8.968	0.000	6.74e+04	1.05e+05
zipcode_98065	1.264e+05	1.49e+04	8.473	0.000	9.72e+04	1.56e+05
zipcode_98070	7.2e+04	1.14e+04	6.334	0.000	4.97e+04	9.43e+04
zipcode_98072	1.227e+05	1.76e+04	6.978	0.000	8.82e+04	1.57e+05
zipcode_98074	1.88e+05	1.42e+04	13.231	0.000	1.6e+05	2.16e+05
zipcode_98075	2.145e+05	1.37e+04	15.671	0.000	1.88e+05	2.41e+05
zipcode_98077	1.269e+05	1.83e+04	6.929	0.000	9.1e+04	1.63e+05
zipcode_98092	-7663.2632	7003.594	-1.094	0.274	-2.14e+04	6064.345
zipcode_98102	3.42e+05	1.55e+04	22.013	0.000	3.12e+05	3.72e+05
zipcode_98103	2.507e+05	1.43e+04	17.591	0.000	2.23e+05	2.79e+05
zipcode_98105	3.089e+05	1.49e+04	20.695	0.000	2.8e+05	3.38e+05
zipcode_98106	7.308e+04	1.05e+04	6.969	0.000	5.25e+04	9.36e+04
zipcode_98107	2.481e+05	1.47e+04	16.910	0.000	2.19e+05	2.77e+05
zipcode_98108	7.486e+04	1.16e+04	6.476	0.000	5.22e+04	9.75e+04
zipcode_98109	3.432e+05	1.56e+04	21.980	0.000	3.13e+05	3.74e+05
zipcode_98112	3.687e+05	1.4e+04	26.416	0.000	3.41e+05	3.96e+05
zipcode_98115	2.535e+05	1.45e+04	17.488	0.000	2.25e+05	2.82e+05
zipcode_98116	2.366e+05	1.18e+04	20.095	0.000	2.13e+05	2.6e+05
zipcode_98117	2.387e+05	1.47e+04	16.278	0.000	2.1e+05	2.67e+05
zipcode_98118	1.236e+05	1.02e+04	12.064	0.000	1.04e+05	1.44e+05
zipcode_98119	3.321e+05	1.45e+04	22.835	0.000	3.04e+05	3.61e+05
zipcode_98122	2.458e+05	1.28e+04	19.265	0.000	2.21e+05	2.71e+05
zipcode_98125	1.291e+05	1.56e+04	8.261	0.000	9.85e+04	1.6e+05
zipcode_98126	1.385e+05	1.08e+04	12.873	0.000	1.17e+05	1.6e+05
zipcode_98133	7.66e+04	1.61e+04	4.744	0.000	4.5e+04	1.08e+05
zipcode_98136	2.047e+05	1.11e+04	18.520	0.000	1.83e+05	2.26e+05
zipcode_98144	1.918e+05	1.19e+04	16.157	0.000	1.68e+05	2.15e+05
zipcode_98146	7.139e+04	9838.756	7.256	0.000	5.21e+04	9.07e+04
zipcode_98148	3.006e+04	1.32e+04	2.273	0.023	4139.841	5.6e+04
zipcode_98155	6.584e+04	1.68e+04	3.919	0.000	3.29e+04	9.88e+04
zipcode_98166	7.007e+04	9010.639	7.777	0.000	5.24e+04	8.77e+04
zipcode_98168	1.879e+04	9482.888	1.981	0.048	201.759	3.74e+04
zipcode_98177	1.401e+05	1.69e+04	8.267	0.000	1.07e+05	1.73e+05
zipcode_98178	2.552e+04	9804.820	2.602	0.009	6298.076	4.47e+04
zipcode_98188	1.038e+04	9994.135	1.039	0.299	-9209.027	3e+04

zipcode_98198	5223.4014	7582.058	0.689	0.491	-9638.043	2.01e+04
zipcode_98199	2.926e+05	1.4e+04	20.843	0.000	2.65e+05	3.2e+05
has_larger_sqft_than_neighbors	-1.718e+04	1719.318	-9.993	0.000	-2.06e+04	-1.38e+04

Omnibus:	1997.283	**Durbin-Watson:**	1.988
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	7373.063
Skew:	0.457	**Prob(JB):**	0.00
Kurtosis:	5.797	**Cond. No.**	291.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.





Even after scaling, the bedrooms coefficient seems to be insignificant. This could be due to the actual sqft mattering more and being a better predictor of sales price. We can go ahead and drop the bedrooms column and take a look at the coefficients for further insight.

In [66]:
```
df_scaled.drop('bedrooms', axis=1, inplace=True)
model = model_lin_reg(df=df_scaled)
```

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.826
Model:	OLS	**Adj. R-squared:**	0.826
Method:	Least Squares	**F-statistic:**	1166.
Date:	Mon, 19 Apr 2021	**Prob (F-statistic):**	0.00
Time:	12:13:22	**Log-Likelihood:**	-2.6138e+05
No. Observations:	20439	**AIC:**	5.229e+05
Df Residuals:	20355	**BIC:**	5.236e+05
Df Model:	83		
Covariance Type:	nonrobust		

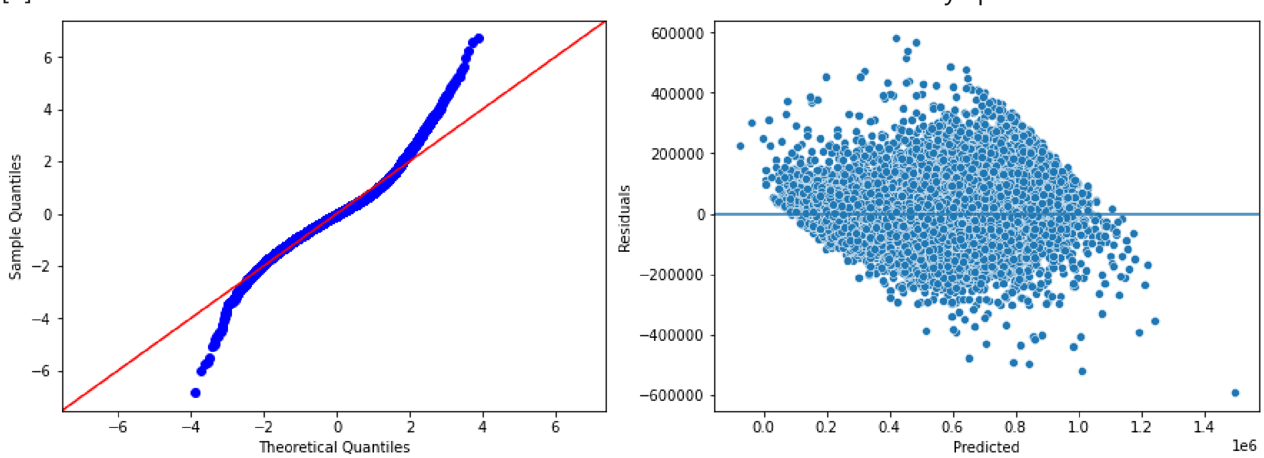
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3.303e+05	1e+04	32.877	0.000	3.11e+05	3.5e+05
bathrooms	1.377e+04	1013.577	13.590	0.000	1.18e+04	1.58e+04
sqft_lot	1.202e+04	667.475	18.011	0.000	1.07e+04	1.33e+04
floors	-1.329e+04	944.113	-14.078	0.000	-1.51e+04	-1.14e+04
waterfront	7489.9003	651.556	11.495	0.000	6212.797	8767.003
view	2.385e+04	672.182	35.482	0.000	2.25e+04	2.52e+04
condition	1.617e+04	690.800	23.406	0.000	1.48e+04	1.75e+04
grade	5.007e+04	1062.780	47.117	0.000	4.8e+04	5.22e+04
sqft_above	9.563e+04	1314.489	72.748	0.000	9.31e+04	9.82e+04
yr_built	-1.667e+04	1051.449	-15.855	0.000	-1.87e+04	-1.46e+04
lat	2.181e+04	4950.717	4.406	0.000	1.21e+04	3.15e+04
long	-9339.4918	3566.940	-2.618	0.009	-1.63e+04	-2348.002
renovated	5791.9948	641.256	9.032	0.000	4535.081	7048.909
has_basement	4.717e+04	1636.087	28.833	0.000	4.4e+04	5.04e+04
zipcode_98002	6079.6824	7819.678	0.777	0.437	-9247.516	2.14e+04
zipcode_98003	-1.128e+04	6990.820	-1.613	0.107	-2.5e+04	2426.563
zipcode_98004	4.749e+05	1.37e+04	34.573	0.000	4.48e+05	5.02e+05
zipcode_98005	2.939e+05	1.39e+04	21.180	0.000	2.67e+05	3.21e+05
zipcode_98006	2.484e+05	1.14e+04	21.768	0.000	2.26e+05	2.71e+05
zipcode_98007	2.176e+05	1.42e+04	15.283	0.000	1.9e+05	2.46e+05
zipcode_98008	2.005e+05	1.36e+04	14.751	0.000	1.74e+05	2.27e+05
zipcode_98010	9.968e+04	1.2e+04	8.313	0.000	7.62e+04	1.23e+05
zipcode_98011	8.508e+04	1.76e+04	4.824	0.000	5.05e+04	1.2e+05
zipcode_98014	8.342e+04	1.94e+04	4.299	0.000	4.54e+04	1.21e+05
zipcode_98019	5.968e+04	1.91e+04	3.126	0.002	2.23e+04	9.71e+04
zipcode_98022	2.404e+04	1.05e+04	2.294	0.022	3502.443	4.46e+04
zipcode_98023	-2.91e+04	6450.225	-4.512	0.000	-4.17e+04	-1.65e+04
zipcode_98024	1.335e+05	1.72e+04	7.771	0.000	9.99e+04	1.67e+05
zipcode_98027	1.777e+05	1.16e+04	15.366	0.000	1.55e+05	2e+05
zipcode_98028	7.239e+04	1.72e+04	4.219	0.000	3.88e+04	1.06e+05
zipcode_98029	2.089e+05	1.32e+04	15.786	0.000	1.83e+05	2.35e+05
zipcode_98030	1409.0238	7704.114	0.183	0.855	-1.37e+04	1.65e+04
zipcode_98031	2269.2724	8043.564	0.282	0.778	-1.35e+04	1.8e+04
zipcode_98032	-1.804e+04	9301.251	-1.939	0.053	-3.63e+04	195.624

zipcode_98033	2.701e+05	1.48e+04	18.203	0.000	2.41e+05	2.99e+05
zipcode_98034	1.25e+05	1.58e+04	7.904	0.000	9.4e+04	1.56e+05
zipcode_98038	5.001e+04	8718.135	5.736	0.000	3.29e+04	6.71e+04
zipcode_98039	6.029e+05	3.75e+04	16.071	0.000	5.29e+05	6.76e+05
zipcode_98040	3.904e+05	1.22e+04	32.021	0.000	3.66e+05	4.14e+05
zipcode_98042	1.043e+04	7403.795	1.408	0.159	-4085.059	2.49e+04
zipcode_98045	1.111e+05	1.62e+04	6.865	0.000	7.94e+04	1.43e+05
zipcode_98052	2.092e+05	1.5e+04	13.927	0.000	1.8e+05	2.39e+05
zipcode_98053	2.012e+05	1.61e+04	12.472	0.000	1.7e+05	2.33e+05
zipcode_98055	2.192e+04	8989.182	2.438	0.015	4297.280	3.95e+04
zipcode_98056	7.812e+04	9814.117	7.960	0.000	5.89e+04	9.74e+04
zipcode_98058	2.411e+04	8505.148	2.834	0.005	7436.354	4.08e+04
zipcode_98059	8.62e+04	9614.416	8.965	0.000	6.74e+04	1.05e+05
zipcode_98065	1.264e+05	1.49e+04	8.474	0.000	9.72e+04	1.56e+05
zipcode_98070	7.211e+04	1.14e+04	6.348	0.000	4.98e+04	9.44e+04
zipcode_98072	1.227e+05	1.76e+04	6.978	0.000	8.82e+04	1.57e+05
zipcode_98074	1.88e+05	1.42e+04	13.230	0.000	1.6e+05	2.16e+05
zipcode_98075	2.145e+05	1.37e+04	15.671	0.000	1.88e+05	2.41e+05
zipcode_98077	1.269e+05	1.83e+04	6.929	0.000	9.1e+04	1.63e+05
zipcode_98092	-7660.1869	7003.428	-1.094	0.274	-2.14e+04	6067.095
zipcode_98102	3.42e+05	1.55e+04	22.018	0.000	3.12e+05	3.72e+05
zipcode_98103	2.507e+05	1.43e+04	17.593	0.000	2.23e+05	2.79e+05
zipcode_98105	3.089e+05	1.49e+04	20.694	0.000	2.8e+05	3.38e+05
zipcode_98106	7.308e+04	1.05e+04	6.970	0.000	5.25e+04	9.36e+04
zipcode_98107	2.482e+05	1.47e+04	16.914	0.000	2.19e+05	2.77e+05
zipcode_98108	7.488e+04	1.16e+04	6.478	0.000	5.22e+04	9.75e+04
zipcode_98109	3.433e+05	1.56e+04	21.986	0.000	3.13e+05	3.74e+05
zipcode_98112	3.688e+05	1.4e+04	26.420	0.000	3.41e+05	3.96e+05
zipcode_98115	2.535e+05	1.45e+04	17.490	0.000	2.25e+05	2.82e+05
zipcode_98116	2.366e+05	1.18e+04	20.103	0.000	2.14e+05	2.6e+05
zipcode_98117	2.387e+05	1.47e+04	16.282	0.000	2.1e+05	2.67e+05
zipcode_98118	1.236e+05	1.02e+04	12.067	0.000	1.04e+05	1.44e+05
zipcode_98119	3.321e+05	1.45e+04	22.841	0.000	3.04e+05	3.61e+05
zipcode_98122	2.459e+05	1.28e+04	19.268	0.000	2.21e+05	2.71e+05
zipcode_98125	1.291e+05	1.56e+04	8.260	0.000	9.85e+04	1.6e+05

zipcode_98126	1.386e+05	1.08e+04	12.884	0.000	1.17e+05	1.6e+05
zipcode_98133	7.658e+04	1.61e+04	4.743	0.000	4.49e+04	1.08e+05
zipcode_98136	2.047e+05	1.1e+04	18.531	0.000	1.83e+05	2.26e+05
zipcode_98144	1.918e+05	1.19e+04	16.160	0.000	1.69e+05	2.15e+05
zipcode_98146	7.139e+04	9838.519	7.257	0.000	5.21e+04	9.07e+04
zipcode_98148	3.009e+04	1.32e+04	2.275	0.023	4163.796	5.6e+04
zipcode_98155	6.579e+04	1.68e+04	3.917	0.000	3.29e+04	9.87e+04
zipcode_98166	7.008e+04	9010.377	7.778	0.000	5.24e+04	8.77e+04
zipcode_98168	1.88e+04	9482.544	1.983	0.047	216.730	3.74e+04
zipcode_98177	1.401e+05	1.69e+04	8.267	0.000	1.07e+05	1.73e+05
zipcode_98178	2.549e+04	9804.095	2.600	0.009	6270.930	4.47e+04
zipcode_98188	1.035e+04	9993.181	1.035	0.301	-9241.852	2.99e+04
zipcode_98198	5236.4108	7581.752	0.691	0.490	-9624.433	2.01e+04
zipcode_98199	2.926e+05	1.4e+04	20.850	0.000	2.65e+05	3.2e+05
has_larger_sqft_than_neighbors	-1.719e+04	1718.780	-10.003	0.000	-2.06e+04	-1.38e+04

Omnibus:	1998.403	**Durbin-Watson:**	1.987
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	7377.212
Skew:	0.458	**Prob(JB):**	0.00
Kurtosis:	5.797	**Cond. No.**	284.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



iNTERPRET

Final Scaled Coefficients

In [67]:
```
coeffs = model.params.sort_values().to_frame('coeffs')
coeffs['abs'] = coeffs['coeffs'].abs()
coeffs.sort_values('abs', ascending=False, inplace=True)
```

In [68]:
```
coeffs[~coeffs.index.str.startswith('zipcode')]
```

Out[68]:

	coeffs	abs
Intercept	330273.764168	330273.764168
sqft_above	95626.887282	95626.887282
grade	50074.835277	50074.835277
has_basement	47173.818402	47173.818402
view	23850.371935	23850.371935
lat	21810.707948	21810.707948
has_larger_sqft_than_neighbors	-17192.309109	17192.309109
yr_built	-16670.283873	16670.283873
condition	16169.003737	16169.003737
bathrooms	13774.717849	13774.717849
floors	-13290.789002	13290.789002
sqft_lot	12021.914526	12021.914526
long	-9339.491807	9339.491807
waterfront	7489.900314	7489.900314
renovated	5791.994762	5791.994762

When compared to each other, we are seeing that the top 3 parameters that affect the sales price of a home are the total sqft above ground (total living sqft, excluding any basements), the grade of construction/finishes as well as whether the house had a basement or not.

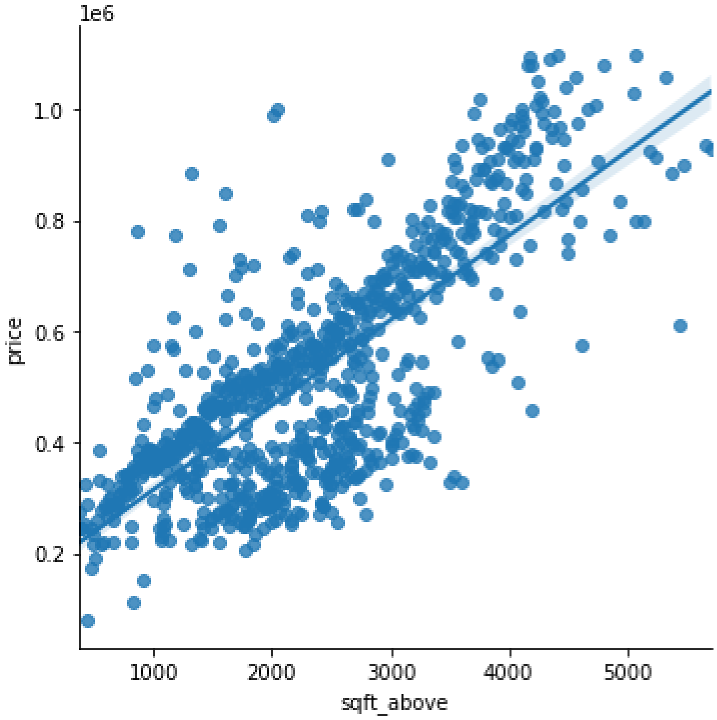
In [69]:
```
df_sqft_above = df_IQR_price.loc[:,['price','sqft_above']]
```

In [70]:
```
df_sqft_above = df_sqft_above.groupby('sqft_above').mean().reset_index()
```

Data Visualizations

In [71]:
```
sns.lmplot(x='sqft_above', y='price', data=df_sqft_above)
```

Out[71]:
```
<seaborn.axisgrid.FacetGrid at 0x1c35ee3e8e0>
```



In [72]:
```
print(df_sqft_above['sqft_above'].min())
print(df_sqft_above['sqft_above'].max())
```

```
370
5710
```

In [73]:
```
def categorize(x):
    if (x<1000) & (x > 0):
        val = 'Up to 1000 SF'
    elif (x>=1000) & (x<2000):
        val = '1000-2000 SF'
    elif (x>=2000) & (x<3000):
        val = '2000-3000 SF'
    elif (x>=3000) & (x<4000):
        val = '3000 - 4000 SF'
    elif (x>=4000) & (x<5000):
        val = '4000 - 5000 SF'
    else:
        val = '5000+ SF'
    return val
```

In [74]:
```
df_sqft_above['Category'] = df_sqft_above['sqft_above'].map(lambda x: categorize(x))
```

In [75]:
```
df_sqft_above['Category'].value_counts()
```

Out[75]:
```
2000-3000 SF      272
1000-2000 SF      263
3000 - 4000 SF    151
Up to 1000 SF      78
4000 - 5000 SF     60
5000+ SF           12
Name: Category, dtype: int64
```

In [76]:
```
mean_price_per_cat = df_sqft_above.groupby('Category')['price'].mean().reset_index()
```

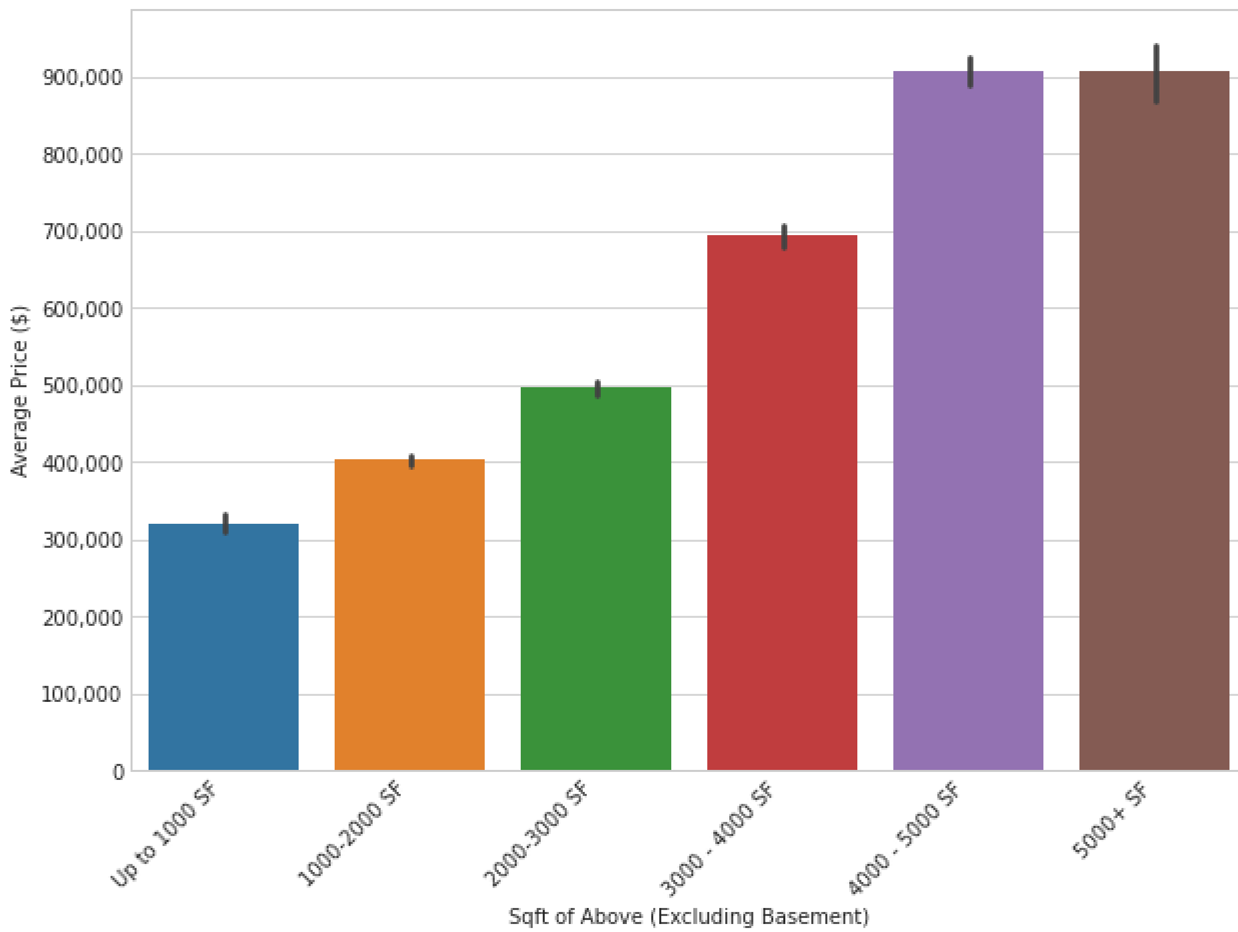
In [77]:
```
mean_price_per_cat
```

Out[77]:

	Category	price
0	1000-2000 SF	403132.561189
1	2000-3000 SF	496939.230583
2	3000 - 4000 SF	694294.614732
3	4000 - 5000 SF	908895.025397
4	5000+ SF	907491.666667
5	Up to 1000 SF	321905.992281

In [78]:

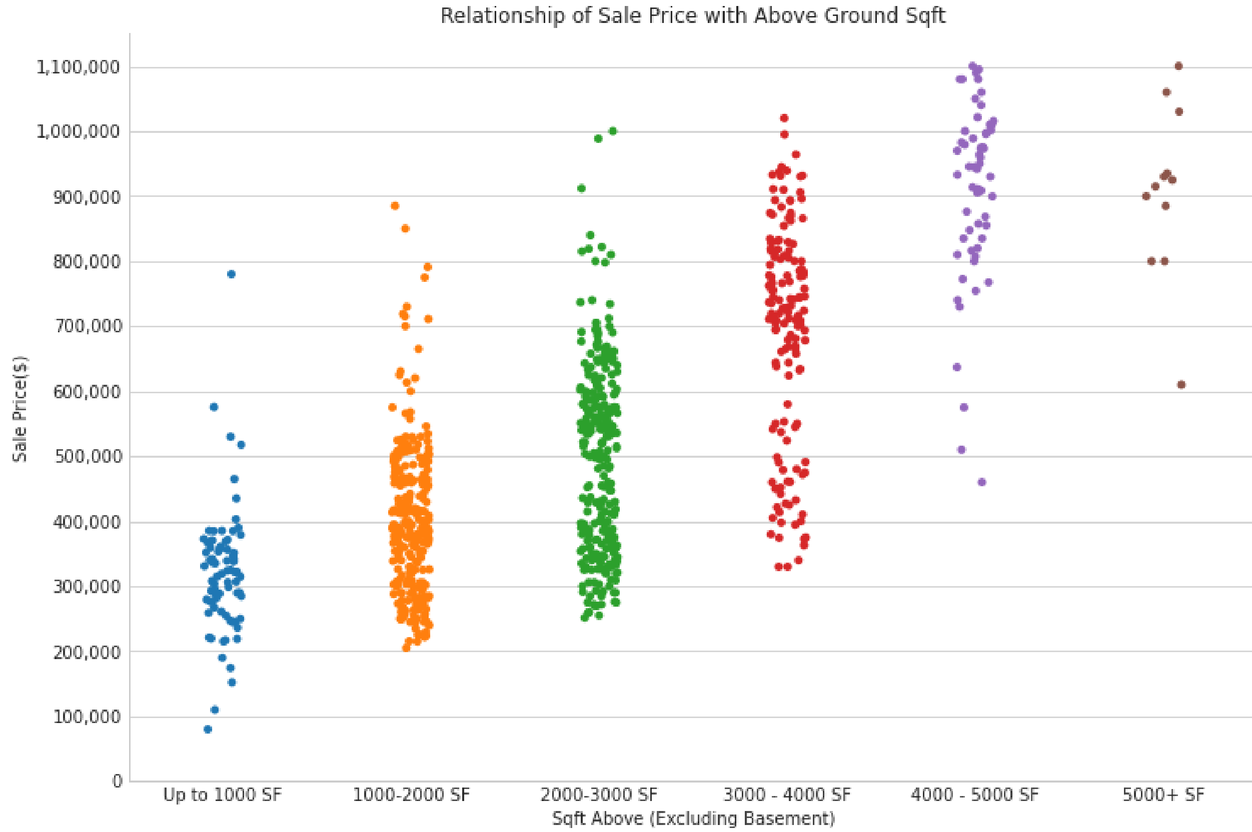
```
from matplotlib.ticker import FuncFormatter
order = ['Up to 1000 SF', '1000-2000 SF', '2000-3000 SF', '3000 - 4000 SF', '4000 - 500
with plt.style.context('seaborn-whitegrid'):
    fig, ax = plt.subplots(figsize=(10,7))
    sns.barplot(data = df_sqft_above, x='Category', y= 'price', order=order, ci=68)
    ax.set_xlabel('Sqft of Above (Excluding Basement)')
    ax.set_ylabel('Average Price ($)')
    ax.yaxis.set_major_formatter(FuncFormatter(lambda x, p: format(int(x), ',')));
    ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
    ax.set_yticks(range(0,1000000,100000))
```



In [79]:

```
with plt.style.context('seaborn-whitegrid'):
    sns.catplot(data = df_sqft_above, x='Category', y='price', aspect=1.5, order=order)
    ax = plt.gca()
    fig = plt.gcf()
    ax.set_yticks(range(0,1200000,100000))
```

```
    ax.yaxis.set_major_formatter(FuncFormatter(lambda x, p: format(int(x), ',')))
    ax.set_ylabel('Sale Price($)')
    fig.set_size_inches(10, 7)
    ax.set_xlabel('Sqft Above (Excluding Basement)')
    ax.set_title('Relationship of Sale Price with Above Ground Sqft')
#     sns.pointplot(x= 'Category', y='price', data=mean_price_per_cat, order=order)
```



As can be seen above, as the square footage of the house increased sale price of the home also tended to increase with it. Even though there is a spread of price at each category of square footage and therefore some overlaps between them, there is a clear positive trend between sale price and square footage above ground. Additionally, it should be noted that as the square footage increases the minimum sale price of the category is consistently increasing as well indicating that even if the house may not be optimal per other parameters, having a certain amount of square footage tends to make it more valuable.

In [80]:

```
df_IQR_price['has_basement'] = df_IQR_price['has_basement'].astype(bool)
```

```
<ipython-input-80-8ecde1b6317e>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_IQR_price['has_basement'] = df_IQR_price['has_basement'].astype(bool)
```

In [81]:

```
from matplotlib.ticker import FuncFormatter
with plt.style.context('seaborn-whitegrid'):

    sns.lmplot(x='sqft_above', y='price', data=df_IQR_price, hue='has_basement',
               aspect=2, scatter_kws=dict(alpha=0.5), palette=sns.color_palette("Spectr
    ax = plt.gca()
    ax.set_xlabel('Sqft Above Ground (Excluding Basement)')
```

```
    ax.set_ylabel('Sale Price ($)')
    ax.set_title("Relationship of Sale Price with Above Ground Sqft and Having a Baseme
    ax.yaxis.set_major_formatter(FuncFormatter(lambda x, p: format(int(x), ',')))
    ax.xaxis.set_ticks(range(0,6500,500))
    ax.set_ylim(0, 1300000)
    plt.gcf().set_size_inches(12,7);
```

When we look at the relationship between square footage above ground and the home's sale price again, but add in a second parameter to define whether the house had a basement or not, the houses with a basement have a slightly more positive relationship with the sale price. So if a house had a basement, it tended to have a slightly higher price than a comparable home. This is visible from the difference of slopes between the two lines shown above where the green line is diverging from the blue line in a positive way.

In [82]:
```
df_grade = df_IQR_price.loc[:,['price','grade']]
df_grade
```

Out[82]:

	price	grade
0	221900.0	7
1	538000.0	7
2	180000.0	6
3	604000.0	7
4	510000.0	8
...	...	...
21592	360000.0	8
21593	400000.0	8
21594	402101.0	7
21595	400000.0	8

	price	grade
21596	325000.0	7

20439 rows × 2 columns

In [83]:
```
def categorize_grade(x):
    if (x<7):
        val = 'Below Average'
    elif (x<=8) & (x>6):
        val = 'Average'
    else:
        val = 'Above Average'

    return val
```

In [84]:
```
df_grade['Category'] = df_grade['grade'].map(lambda x: categorize_grade(x))
```

In [85]:
```
df_grade['Category'].value_counts()
```

Out[85]:
```
Average          14905
Above Average     3227
Below Average     2307
Name: Category, dtype: int64
```

In [86]:
```
df_grade
df_grade_sqft = pd.concat([df_grade, pd.DataFrame(df_IQR_price['sqft_above'])], axis=1)
df_grade_sqft
```

Out[86]:

	price	grade	Category	sqft_above
0	221900.0	7	Average	1180
1	538000.0	7	Average	2170
2	180000.0	6	Below Average	770
3	604000.0	7	Average	1050
4	510000.0	8	Average	1680
...	...	...	...	...
21592	360000.0	8	Average	1530
21593	400000.0	8	Average	2310
21594	402101.0	7	Average	1020
21595	400000.0	8	Average	1600
21596	325000.0	7	Average	1020

20439 rows × 4 columns

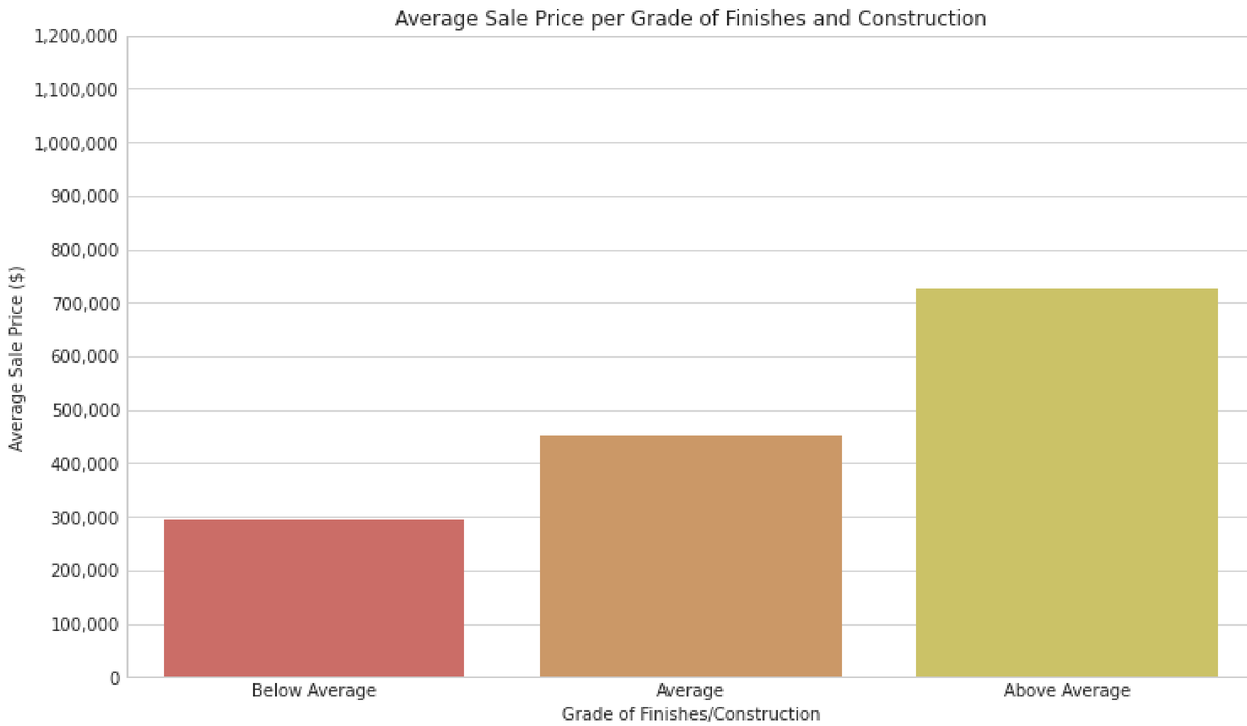
In [87]:
```
mean_price_per_grade = df_grade.groupby('Category')['price'].mean().reset_index()
mean_price_per_grade
```

Out[87]:

	Category	price

	Category	price
0	Above Average	726319.069724
1	Average	450805.678497
2	Below Average	294856.879931

In [88]:
```
with plt.style.context('seaborn-whitegrid'):
    order=['Below Average', 'Average', 'Above Average']
    fig, ax = plt.subplots(figsize=(12,7))
    sns.barplot(data = mean_price_per_grade, x='Category', y= 'price', order=order,
                palette=sns.color_palette("hls", 14))
    ax.set_xlabel('Grade of Finishes/Construction')
    ax.set_ylabel('Average Sale Price ($)')
    ax.set_title('Average Sale Price per Grade of Finishes and Construction')
    ax.set_ylim(0,1200000)
    ax.yaxis.set_major_formatter(FuncFormatter(lambda x, p: format(int(x), ',')));
    ax.set_yticks(range(0,1300000,100000))
```



Another interesting, but also expected relationship is between the grade of finishes and the sale price. Houses that had higher grades of finishes and a better construction quality sold for higher prices.

In [89]:
```
from matplotlib.ticker import FuncFormatter
with plt.style.context('seaborn-whitegrid'):

    sns.lmplot(x='sqft_above', y='price', data=df_grade_sqft, hue='Category',
               aspect=2, scatter_kws=dict(alpha=0.7), palette=sns.color_palette("hls",
    ax = plt.gca()
    ax.set_xlabel('Sqft Above Ground (Excluding Basement)')
    ax.set_ylabel('Sale Price ($)')
    ax.set_title("Relationship of Sale Price with Above Ground Sqft and Grade")
    ax.yaxis.set_major_formatter(FuncFormatter(lambda x, p: format(int(x), ',')))
    ax.xaxis.set_ticks(range(0,6500,500))
```

```
    ax.set_ylim(0, 1300000)
    plt.gcf().set_size_inches(12,7);
```

When grade is plotted with square footage above ground, the relationship shown in the previous visual becomes even more apparent. As can be seen from the different colored data points, above average homes tended to have a higher sale price. The regression lines for average and above average have a higher positive slope compared to the below average homes, which means that as the square footage of a home increases, the higher graded homes will tend to have higher prices.

CONCLUSIONS & RECOMMENDATIONS

Even though renovations are usually a lot of effort and stressful to lots of homeowners, they may help increase the property's value. To sum up, our analysis for King County, Washington showed the following:

- Increasing the square footage above ground tends to increase the house's value.
- Focusing on the grade of finishes and the quality of construction as a whole tends to pay dividends when it comes to selling the house.
- Having a basement is the third most effective parameter in increasing a home's sale price.

Our first recommendation based on our findings above would be for the homeowner to renovate their house and add livable above ground square footage to the property. This could be in the shape of an added extension to the home or a simpler approach of finishing the attic space.

Secondly, we would advise the homeowner to also focus their renovation efforts on the finishes and quality of the materials that they would be choosing to use. This could include components such as kitchen countertop material, cabinetry design, bathroom fixtures, general lighting design and fixture choices inside and outside the house, flooring.

Lastly, we would propose that the homeowner add a basement to their home. It should be noted that since we do not know what the construction costs of this undertaking would be like versus the gain in the home's sale price, it is difficult to predict the outcome (refer to the Limitations & Next Steps section for more information on this). However, based on our model, we found that having a basement for homes was the third most effective parameter. If the home already has a basement, but the basement is unfinished, our recommendation would be for the homeowner to make the basement an occupiable space by finishing it.

Limitations & Next Steps

Given more time and information about what the homeowner's renovation budget would be, we would have wanted to analyze whether these top 3 parameters would truly be the most effective in bringing a net value increase since a renovation such as adding a basement to a home would be very costly and may not end up returning a net value increase. Additionally, the construction costs in the state of Washington may be higher than other states due to factors such as permitting, material costs, logistical challenges etc. which may effect the net value increase as well. Furthermore, having information about whether the homeowner is thinking about living in the renovated house or renting it out would allow us to fine tune our analysis and bring more valuable insight.